AN ABSTRACT REPRESENTATION AND ANALYSIS OF PRODUCTION LINES WITH INTER-STAGE STORAGE

by

Kin Ming Chan, B.Eng., M.Eng.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Doctor of Philosophy

McMaster University

AN ABSTRACT REPRESENTATION AND ANALYSIS OF PRODUCTION LINES WITH INTER-STAGE STORAGE

DOCTOR OF PHILOSOPHY (1994) McMaster University
(Electrical Engineering) Hamilton, Ontario

TITLE:	An Abstract Representation And Analysis of Production Lines With Inter-Stage Storage
AUTHOR:	Kin Ming Chan, B.End., M.Eng. (McMaster University)
SUPERVISOR:	Dr. Barna Szabados, Department of Electrical and Computer Engineering
NUMBER OF PAGES:	xxiv, 230

ABSTRACT

This thesis presents a novel methodology to represent a production system with inter-stage storages and variable cycle times. The 'abstract representation', which employs no approximation and is not based on probability, involves: (1) completely representing a system with stages and storages by elements only, each of which contains one stage and some storages, (2) measuring each element independently by supplying a predefined probing rate of job flow across an element and (3) formulating the instantaneous relationships between the independent measurements and system parameters.

This 'abstract representation' is then illustrated by three different problems. The first problem involves on-line system monitoring which requires locating the origins of production loss in a system in real-time. The 'abstract representation' is used to derive the instantaneous relationships between the causes and effects of system production loss among the elements to address this problem.

After the instantaneous relationships between the causes and effects of production loss are derived, a natural extension of this is to apply these relationships to address the problems concerning system performance improvements. Therefore, this thesis will address the second and third problems which are system production control and storage allocation.

The objectives of production control as defined in (Ryzin et al. 93) are to control the system so that one job will be produced in one unit time and to minimize the inventory in the (infinite) inter-stage storage. In this thesis, an algorithm will be derived to address the problem of production control. This algorithm is based on the instantaneous relationships between the causes and effects of production loss and will be designed to theoretically accommodate systems of any size with arbitrary cycle time behaviours. The storage capacities of the system are finite and can have any value. Therefore, the minimization of inventory being carried in an infinite storage is not part of the objective of the production control algorithm in this thesis. This production control algorithm will calculate the maximum allowable cycle time for each stage for each cycle, so that if these maximum allowable cycle times are met by the stages, the goal of production control can be achieved. Furthermore, the maximum allowable cycle times calculated by the production control algorithm in this thesis will be bounded below so that the maximum cycle time values should fall within an acceptable range.

The goal of storage allocation in a system is to find the amount of storage space between the stages which yields the maximum system efficiency for a certain type of cycle time distribution while satisfying a set of pre-defined constraints. In this thesis, a preliminary study will be undertaken for this problem with the constraint being that the total available storage space is fixed. This preliminary study tries to show that the approximated long term relationships between the causes and effects of production loss can be helpful in providing insights which help solve the problem of storage allocation.

ACKNOWLEDGEMENTS

I would like to express my thanks and appreciation to my supervisor, Dr. Barna Szabados for his invaluable insight, encouragement, patience and open mindedness throughout the course of this work.

Special thanks to Mr. Orest Storoshchuk of General Motors of Canada for his many valuable suggestions, one of them being the topic of this thesis and Mr. Payman Hodaie and Mr. Gary Parkinson also of General Motors of Canada for their knowledgable discussion and help on the computer system for real-time data collection.

I would also like to thank my fellow graduate students, especially Dr. Sallehhudin Bin Yusof and Mr. Kyle Mckay for being so supportive and for making my studies an enjoyable and memorable experience.

Thanks to my parents for their love, wisdom and continuing support.

Finally, I would like to thank my dear wife, Joanne, for her many hours spent proof-reading this thesis as well as for her love and untiring support.

TABLE OF CONTENTS

ABSTRACT iii

ACKNOWLEDGEMENTS v

TABLE OF CONTENTS vi

NOMENCLATURE xiv

LIST OF FIGURES xix

CHAPTER 1

INTRODUCTION 1

1.1 INTRODUCTION 1

1.1.1 Assembly System with Inter-Stage Storages 2

1.1.2 System Availability of Assembly Systems with Inter-Stage Storages 3

1.1.3 Line Balancing and Assembly Systems with Inter-Stage Storages 4

1.2 BACKGROUND 5

1.3 THE PROPOSED APPROACH 7

1.4 THESIS STRUCTURE 8

1.5 CONCLUSION 9

CHAPTER 2

SURVEY OF RELEVANT APPROACHES 11

2.1 INTRODUCTION 11

2.2 ANALYTICAL APPROACH 12

2.2.1 Markov Process 12

2.2.2 Finite Queues In Series 13

2.2.3 Random Breakdowns and Repairs 14

2.3 SIMULATION APPROACH 21

2.4 HEURISTIC APPROACH 22

2.5 OPTIMAL STORAGE ALLOCATION 24

2.6 PRODUCTION CONTROL 28

2.7 CONCLUSION 30

CHAPTER 3

OVERVIEW OF THE PROPOSED METHODOLOGY 32

3.1 INTRODUCTION 32

3.2 THE PROBLEM 32

3.3 ASSUMPTIONS 35

3.4 THE ABSTRACTION 37

3.5 THESIS OBJECTIVES 39

3.6 CONCLUSION 40

CHAPTER 4

THE INDEPENDENT MEASUREMENT 44

4.1 INTRODUCTION 44

4.1.1 The Ideal World Definition 44

4.1.2 The Ideal World Measurement 46

4.2 IDEAL WORLD MEASUREMENT DERIVATION 47

4.2.1 Ideal World Output Delay 49

4.2.2 Ideal World Output Wait 52

4.2.3 Ideal World Input Delay 54

4.2.4 Ideal World Input Wait 55

4.3 CHANGES IN IDEAL WORLD STORAGE CONTENT WHEN WAITING 58

4.3.1 Ideal World Input Wait and Output Storage Content 58

4.3.2 Ideal World Output Wait and Input Storage Content 60

4.4 IDEAL WORLD OPERATION TIME 61

4.5 CONCLUSION 62

CHAPTER 5

CONSTRUCTING THE REAL WORLD ENVIRONMENT 65

5.1 INTRODUCTION 65

5.1.1 Notation Conventions 66

5.1.2 The 'Shifting' of an Element 67

5.1.3 The Stand Alone Operation Time of an Ideal World 68

5.2 REAL WORLD OUTPUT WAIT 70

5.2.1 Condition for Real World Output Wait 73

5.2.2 Expressing $iE_{i+1}_late_{kj}$ In Terms of Ideal World Measurements 77

5.3 REAL WORLD INPUT WAIT 79

5.3.1 Condition for Real World Input Wait 81

5.3.2 Expressing $iE_{i-1}_late_{hj}$ in Terms of Ideal World Measurements 85

5.4 REAL WORLD PROPAGATED OUTPUT WAIT 87

5.5 PERFORMANCES 88

5.5.1 Ideal World Performance 88

5.5.2 Real World Performance 90

5.6 CONCLUSION 91

CHAPTER 6

CAUSES AND EFFECTS OF PRODUCTION LOSS 96

6.1 INTRODUCTION 96

6.2 The Causes of Real World Waits Between Two Elements 96

6.2.1 Real World Output Wait Causes 97

6.2.2 Real World Input Wait Causes 99

6.3 CAUSES AND EFFECTS OF REAL WORLD WAITS 100

6.3.1 System Monitoring 105

6.4 EMULATION EXAMPLES 106

6.4.1 Summary of Emulation Examples 111

6.5 RELATIONSHIPS BETWEEN REAL WORLD WAITS AND ELEMENT SHIFTING 113

6.5.1 Summary 118

6.6 SOFTWARE IMPLEMENTATION 120

6.7 CONCLUSION 121

CHAPTER 7

APPLICATION EXAMPLES 124

7.1 INTRODUCTION 124

7.2 EXAMPLE I: THE PRODUCTION CONTROL EXAMPLE 124

7.2.1 The Maximum Cycle Time Algorithm 126

7.2.1.1 The Preliminary Criterion 127

7.2.1.2 The Criterion 128

7.2.1.2.1 Argument for The Criterion 129

7.2.1.2.2 Analysing the Criterion 132

7.2.1.2.3 Satisfying the Lower Bound of Maximum Cycle Time Constraint 136

7.2.1.2.4 The Output Stage Under the Criterion . . . 137

7.2.1.2.5 Effect of Propagated Output Wait on the Criterion 139

7.2.1.3 Summary 142

7.2.2 Emulation Examples 142

7.2.2.1 A Three-Stage System Example 143

7.2.2.2 Discussion of The Three-Stage Example 149

7.2.2.3 Ten-Stage System Examples 151

7.2.3 The Minimization Factor 153

7.3 EXAMPLE II: A PRELIMINARY INVESTIGATION OF Storage ALLOCATION 157

7.3.1 Long Term Ideal World Measurement 158

7.3.2 The Storage Allocation Argument 160

7.3.3 The Output Stage Under Storage Allocation 162

7.3.4 Summary of the Storage Allocation Method 164

7.4 CONCLUSION 166

CHAPTER 8

PARALLEL CONFIGURATIONS 170

8.1 INTRODUCTION 170

8.2 JOINT TYPE I 171

8.2.1 The Splitting Element Type I 173

8.2.2 Ideal World Measurement for Splitting Element Type I . . . 174

8.2.3 The Merging Element Type I 176

8.2.4 Ideal World Measurement for Merging Element Type I . . . 176

8.3 JOINT TYPE II 177

8.3.1 The Splitting Element Type II 179

8.3.2 Ideal World Measurement for Splitting Element Type II . . . 180

8.3.3 The Merging Element Type II 183

8.3.4 Ideal World Measurement for Merging Element Type II . . . 184

8.4 REAL WORLD PARAMETERS CALCULATION FOR PARALLEL CONFIGURATIONS 186

8.4.1 Blocking Between Parallel Lanes 187

8.5 SOFTWARE IMPLEMENTATION CONSIDERATIONS 189

8.6 CONCLUSION 190

CHAPTER 9

CONCLUSION 192

9.1 SUMMARY 192

9.2 DISCUSSION 193

9.3 CONTRIBUTION AND CONCLUSION 194

9.4 RECOMMENDATIONS FOR FUTURE RESEARCH 196

APPENDIX A System Output 'Shifted' Plots for a ten-stage system 197

APPENDIX B System Output 'Shifted' Plots after applying the maximum cycle time 'algorithm', MF = 1.00 200

APPENDIX C System Output 'Shifted' Plots after applying the maximum cycle time 'algorithm', MF = 0.75 204

APPENDIX D System Output 'Shifted' Plots after applying the maximum cycle time 'algorithm', MF = 0.5 208

APPENDIX E System Output 'Shifted' Plots after applying the maximum cycle time 'algorithm', MF = 0.25 212

APPENDIX F System Output 'Shifted' Plots after applying the maximum cycle time 'algorithm', MF = 0 (the preliminary criteria) 216

APPENDIX G Figures for the Preliminary Study of Buffer Allocation 219

APPENDIX H Figures for the reversibility property example 221

REFERENCES 223

BIBLIOGRAPHY 226

NOMENCLATURE

The following definitions will be used throughout the thesis:

Emulation: The process of using computer programs to artificially generate all of the necessary parameters for this study. The logical relationships among these generated parameters are the same as those of the real parameters in a real production system, given the assumptions of this study. Alternatively, these parameters can be gathered directly from a real production system.

ct - a cycle time of a stage.

mct - the mean cycle time of a stage.

Sct - the start cycle time of a stage.

Ect - the end cycle time of a stage.

N_i - the (integer) number of job produced by $stage_i$.

real_world_input_wait - the duration of time a stage waited for a job from its upstream stage.

real_world_output_wait - the duration of time a stage waited for an output space for a finished job in front of its downstream stage.

starvation - real world input wait between any two neighbouring stages.

blocking - real world output wait between any two neighbouring stages.

propagated_output_wait - real world output wait among a series of consecutive stages.

storage content - the jobs in the storages between two neighbouring stages.

real world transfer_time - the time required for a job to travel between two neighbouring stages.

element - The name of the entity used to represent a stage and some of its surrounding storages in the system in the abstract representation.

Eff - The efficiency of an element in the system.

splitting joint - the connecting point where jobs flow from a serial configuration to a parallel configuration.

merging joint - the connecting point where jobs flow from a parallel configuration to a serial configuration.

splitting element - The name of the entity used to represent the stage located next to a splitting joint on the serial configuration and some of its surrounding storages in the system in the abstract representation.

merging element - The name of the entity used to represent the stage located next to a merging joint on the serial configuration and some of its surrounding storages in the system in the abstract representation.

The following definitions are used to define an 'ideal world'. The 'ideal world' will be introduced in chapter four.

ideal world - A self-sufficient hypothetical environment for an element, for the purpose of measuring the element independently. It will be defined in chapter four.

ideal world input - A device in the ideal world. It supplies a job flow at a pre-defined rate and has an infinite source of jobs inside.

ideal world output - A device in the ideal world. It demands a job flow at a pre-defined rate and has infinite storage space inside.

stand alone operation time - the operation time of an 'ideal world input' or 'ideal world output' when it is not connected to an element.

N_{out} - the number of jobs received by the 'ideal world output'.

N_{in} - the number of jobs supplied by the 'ideal world input'.

ob_limit - the total storage space in an element between the stage and the 'ideal world output'.

ib_limit - the total input storage space in an element between the 'ideal world input' and the stage plus the space occupied by the stage.

storage sizes - ob_limit, ib_limit.

output_transfer_time - the time required for a job to travel from the stage in an element to the 'ideal world output'.

input_transfer_time - the time required for a job to travel from the 'ideal world input' to the stage in an element.

The following definitions will be used throughout this thesis only for an element under

the ideal world:

The ideal world status parameters:

ib - the jobs in (ib_limit) plus the job currently being processed by the stage of an element.

ob - the jobs in (ob_limit) plus the job currently being processed by the 'ideal world output'.

output storage contents - the jobs inside the storage between the stage and the 'ideal world output'.

input storage contents - the jobs inside the storage between the 'ideal world output' and the stage.

The ideal world measurements:

input_delay - the duration of time the 'ideal world input' waited for a space in the input storage of an element to transfer a job.

output_delay - the duration of time the 'ideal world output' waited for a job from a stage of an element.

input_wait - the duration of time a stage waited for a job from its 'ideal world input'.

output_wait - the duration of time a stage waited for an output space in front of its 'ideal world output'.

ob_empty_time - the time it takes for the output storage of an element to become empty and the 'ideal world output' to become ready to receive a job since the current start cycle time.

ob_fall_time - the time it takes for the output storage of an element to change

from full to having just enough space for one job since the current start cycle time.

ib_full_time - the time it takes for the input storage of an element to become full and the 'input ideal world' to become ready to send out a job since the current start cycle time.

ib_rise_time - the time it takes for the input storage of an element to change from empty to having one job since the current start cycle time.

eff - the efficiency of an element in the ideal world.

A convention will be introduced in chapter five to differentiate between the variables, such as input wait and output wait, from the ideal world to the real world and to distinguish variables among elements. Variable names will be enclosed by parenthesis inside the text.

The cumulative value of any variable is added with 'total' in front of the variable name. For example, total_output_delay is the cumulative value of output_delay. A cumulative value of any variable is accumulated unconditionally since time zero and has the initial value of zero.

LIST OF FIGURES

Figure 1.1 A n-stage production line with inter-stage storage 2

Figure 2.1 A set of 3 two-stage systems for the decomposition of a four-stage system 19

Figure 2.2 Mean deviations from mean production rates at various storage capacities 25

Figure 3.1 A four stage system and its abstract representation 38

Figure 3.2 Block diagram for the proposed abstract representation and analysis 43

Figure 4.1 An element being measured by an "ideal world" 45

Figure 4.2 The content of an element and the "ideal world" measurement 45

Figure 4.3 Illustration of ideal world output delay 50

Figure 4.4 illustration of ideal world input wait 56

Figure 4.5 Illustration of overlap between input wait and output delay 59

Figure 4.6 Illustration of overlap between input wait and output delay, II 60

Figure 4.7 Block diagram for the ideal world calculation procedure 64

Figure 5.1 Conventions for a two element sub-system 67

Figure 5.2 Illustration of the 'shifting' of an Element 67

Figure 5.3 Stand alone ideal world output operation 69

Figure 5.4 Ideal world output operation when connected to an element 69

Figure 5.5 Real World Output World 72

Figure 5.6 Regions of output/input buffer full for $element_i$/$element_{i+1}$, respectively 74

Figure 5.7 Job counts and initial conditions for $element_i$ and $element_{i+1}$ 75

Figure 5.8 Real World Input Wait and Ideal World Measurement 80

Figure 5.9 Regions of output/input buffer full for $element_i$/$element_{i+1}$, respectively 82

Figure 5.10 Block diagram for the real world parameter calculation procedure . . 93

Figure 6.1 Cumulative ideal world delays vs. number of jobs 107

Figure 6.2 Cumulative ideal world delays vs. number of jobs 107

Figure 6.3 Itemized amount of $element_3$ 'shifted' vs. number of jobs 108

Figure 6.4 Cumulative ideal world delays vs. number of jobs 109

Figure 6.5 Cumulative ideal world delays vs. number of jobs 110

Figure 6.6 Itemized amount of $element_3$ 'shifted' vs. number of jobs 110

Figure 6.7 The classes of the object of an element 120

Figure 6.8 The sub-classes of the ideal world class 121

Figure 6.9 Block diagram for the real world parameter calculation and system monitoring procedures 123

Figure 7.1 Cycle time probability density function 145

Figure 7.2 E_3_shifted^$_i$ i=1..3, vs. number of jobs before applying the MCT algorithm 146

Figure 7.3 E_3_shifted^$_i$ i=1..3, vs. number of jobs after applying the MCT algorithm 147

Figure 7.4 E_3_total_output_delay vs. number of jobs after applying the MCT algorithm 148

Figure 7.5 iE_1_total_output_delay and iE_2_total_input_delay after the application of the MCT algorithm 150

Figure 7.6 iE_2_total_output_delay and iE_3_total_input_delay after the application of the MCT algorithm 151

Figure 7.7 Total ideal world delays vs. number of jobs 159

Figure 7.8 Approximate total ideal world delays for ten storage sizes 160

Figure 7.9 Approximate total ideal world delays for different cycle time deviations 161

Figure 7.10 Block diagram for the maximum cycle time algorithm 169

Figure 8.1 Two joints connecting a parallel section to two serial sections 171

Figure 8.2 The type I splitting and merging joints 171

Figure 8.3 The abstract representation of the configuration shown in figure 8.2 . 172

Figure 8.4 The ideal world input and ideal world outputs of a splitting element . 174

Figure 8.5 Ideal world measurement setup for a splitting element with a selected branch 175

Figure 8.6 The type II splitting and merging joints 178

Figure 8.7 The abstract representation of the configuration shown in figure 8.6 . 179

Figure 8.8 The ideal form of splitting element type II 180

Figure 8.9 The ideal form of merging element type II 184

Figure 8.10 Example of a paraliel lane being blocked by another parallel lane . . 187

Figure 8.11 An hypothetical connection between two parallel lanes 188

Figure A.1 E_{10}_shifted^$_i$ i=1..3, vs. number of jobs before applying the MCT algorithm 197

Figure A.2 E_{10}_shifted^$_i$ i=4..7, vs. number of jobs before applying the MCT algorithm 198

Figure A.3 E_{10}_shifted^$_i$ i=8..10, vs. number of jobs before applying the MCT algorithm 198

Figure B.1 E_{10}_shifted^$_i$ i=1..3, vs. number of jobs after applying the MCT algorithm, MF = 1.00 200

Figure B.2 E_{10}_shifted^$_i$ i=4..7, vs. number of jobs after applying the MCT algorithm, MF = 1.00 201

Figure B.3 E_{10}_shifted^$_i$ i=8..10, vs. number of jobs after applying the MCT algorithm, MF = 1.00 201

Figure B.4 iE_{10}_total_output_delay vs. number of jobs after applying the MCT algorithm, MF = 1.00 202

Figure C.1 E_{10}_shifted^$_i$ i=1..3, vs. number of jobs after applying the MCT algorithm, MF = 0.75 204

Figure C.2 E_{10}_shifted^$_i$ i=4..7, vs. number of jobs after applying the MCT algorithm, MF = 0.75 205

Figure C.3 E_{10}_shifted^$_i$ i=8..10, vs. number of jobs after applying the MCT algorithm, MF = 0.75 205

Figure C.4 iE_{10}_total_output_delay vs. number of jobs after applying the MCT algorithm, MF = 0.75 206

Figure D.1 E_{10}_shifted^$_i$ i=1..3, vs. number of jobs after applying the MCT algorithm, MF = 0.50 208

Figure D.2 E_{10}_shifted^$_i$ i=4..7, vs. number of jobs after applying the MCT algorithm, MF = 0.50 209

Figure D.3 E_{10}_shifted^$_i$ i=8..10, vs. number of jobs after applying the MCT algorithm, MF = 0.50 209

Figure D.4 iE_{10}_total_output_delay vs. number of jobs after applying the MCT algorithm, MF = 0.50 210

Figure E.1 E_{10}_shifted^$_i$ i=1..3, vs. number of jobs after applying the MCT algorithm, MF = 0.25 212

Figure E.2 E_{10}_shifted^$_i$ i=4..7, vs. number of jobs after applying the MCT algorithm, MF = 0.25 213

Figure E.3 E_{10}_shifted^$_i$ i=8..10, vs. number of jobs after applying the MCT algorithm, MF = 0.25 213

Figure E.4 iE_{10}_total_output_delay vs. number of jobs after applying the MCT algorithm, MF = 0.25 214

Figure F.1 E_{10}_shifted^$_i$ i=1..3, vs. number of jobs after applying the MCT

algorithm, MF = 0.00 216

Figure F.2 E_{10}_shifted^$_i$ i=4..7, vs. number of jobs after applying the MCT algorithm, MF = 0.00 217

Figure F.3 E_{10}_shifted^$_i$ i=8..10, vs. number of jobs after applying the MCT algorithm, MF = 0.00 217

Figure G.1 Ideal world total waits vs. different storage sizes 220

Figure G.2 Ideal world total waits vs. different cycle time distributions 220

Figure H.1 Itemized E_3_shifted^ 222

Figure H.2 Itemized E_3_shifted^, with $stage_1$ and $stage_3$ interchanged 222

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Assembly lines have been a part of large scale manufacturing systems since they were first introduced to the automobile industry early this century. Traditional assembly lines use a single conveyor mechanism such as a chain, to move jobs along at a constant speed. Operators have to move along with the jobs and return to their starting positions once the jobs are finished. The time available to an operator to perform each job is a constant. The sequence of jobs is maintained throughout the system in such a way that the repair of defective items has to be performed at the end of the line (Buzacott 90).

A traditional assembly line is a synchronous system because the start and end times of all jobs occur at the same instant. The speed of a traditional assembly line is equal to that of the slowest worker if the workers can issue a stop; otherwise the speed is fixed. A fixed speed assembly line can result in there being unfinished products. Other disadvantages of traditional assembly lines include inflexibility, which hinders its utilization and the monotonous nature of jobs provided to the workers, which decreases worker performance and product quality (Buzacott 90). That study also showed that for traditional assembly lines, high quality can only be achieved when utilization of the assembly line is relatively low. Two approaches were evident in the solution of the

problem. One was to replace human workers with machines since machines can perform the same task over a long period of time with a relatively consistent performance. The other one was to modify the existing assembly line to better fit the nature of the human workers. Different methodologies and variations on how to improve the structure of the system were suggested. Among the many methodologies reviewed for today's manufacturing environment (Huang et al. 85), much attention has been paid to assembly systems with inter-stage storages using asynchronous transfer, which is a direct improvement over the traditional assembly lines. Assembly systems with inter-stage storages have been applied to stages employing machines or human workers in high volume manufacturing settings.

1.1.1 Assembly System with Inter-Stage Storages

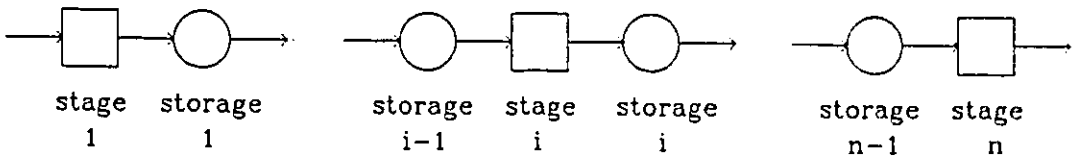


Figure 1.1 A n-stage production line with inter-stage storage

The ideal effect of using inter-stage storages is to 'de-couple' the stages in a system, so that a stage, once it finishes a job, can start on a new job from its input storage, independent of its upstream stage, and the finished job can be sent to its output storage independent of its downstream stage. The available time for each job at each stage can be varied and this flexibility enhances the utilization of each stage (Buzacott 90). The advantages of using inter-stage storages are analytically shown in (Buzacott

66), (Buzacott 71), (Blumenfeld 89).

However, the size of the inter-stage storages is usually limited because of the physical limitations of a factory's floor space in accommodating the inter-stage storages and the non-negligible cost of its installation and maintenance. Therefore the improvements which inter stage storages can provide are also limited. Two observable limitations to the improvements which inter-stage storages can provide are commonly known as the 'blocking' and 'starvation' of a stage. The 'blocking' of a stage occurs when the stage, after it finishes a job and its output storage is full, has to wait until its immediate downstream stage unloads a job from its input storage. The 'starvation' of a stage occurs when the stage, after it finishes a job and its input storage is empty, has to wait until its immediate upstream stage releases a job. The 'de-coupling' effect on inter-stage storages in either of these circumstances is reduced and synchronization between stages as existed with traditional assembly lines is forced locally.

1.1.2 System Availability of Assembly Systems with Inter-Stage Storages

In (J. Masso et al. 74), the system availability of assembly systems with inter-stage storages is formulated for a three stage system. Each stage in the system consists of a machine. The machines all work at equal rates but are subject to stochastic failures. Where there are no inter-stage storages, the percentage of time the system can be expected to be in an operable condition is equal to the percentage of time all three stages are operating. Assuming that the times between failure for the three stages are independent, the expected system availability $E[(P_A)_{SYS}]$ is:

$$E[(P_A)^0_{SYS}] = \prod_{m=1}^{m=3} (P_A)_m \qquad 1.1$$

where the availability of the m^{th} stage is defined as:

$$(P_A)_m = \frac{\text{Mean Time Between Failure}}{\text{Mean Time Between Failure} + \text{Mean Repair Time}} \qquad 1.2$$

With infinite inter-stage storages, a complete 'de-coupling' is achieved among the stages. The theoretical system availability is:

$$(P_A)^*_{SYS} = \text{Min}[(P_A)_1,(P_A)_2,(P_A)_3] \qquad 1.3$$

The allocation of inter-stage storages will serve to increase the system availability from that of the expected value $E[(P_A)_{SYS}]$ towards that of the theoretical maximal value $(P_A)^*_{SYS}$. Intuitively, this result could be applied generally to systems which have any number of stages.

1.1.3 Line Balancing and Assembly Systems with Inter-Stage Storages

Line balancing is one of the factors of assembly system design. An assembly system is balanced if the mean cycle times of all of the stages are equal. This is defined as 'notional' line balancing in (Buxey et al. 73). In (Muth 73) and (Buzacott 71) it is shown that inter-stage storages cannot reduce inefficiencies caused by the unequal mean cycle times of different stages. After a sufficiently long production time, inventory levels of inter-stage storages will either be empty or full in assembly systems which have unequal mean cycle times. Inter-stage storages are useful only if their inventory levels change as production takes place in a balanced system. Therefore, a production system

is usually assumed to be balanced when analysing the effect the inter-stage storage has on it. In a balanced system, system efficiency in terms of system output and idle time is believed to be a function of the number of stages in the system, the cycle time distribution variability and the inter-stage storage capacity (Buxey et al. 72).

Unbalanced systems are studied by (Murphy 78), among others. Unbalancing a balanced system in order to achieve a slightly higher system efficiency is mentioned in (Hillier & Boling 66), among others. The current study as well, examines this conjecture in chapter six.

1.2 BACKGROUND

According to (Gershwin 83), transfer lines were first studied analytically through a probabilistic approach by (Vladzievskii 52). Various approaches used in existing studies are mostly based on probabilities. These approaches can be roughly summarized into the analytical, the simulation and the heuristic approaches. The results provided by the existing studies generally include the steady state system mean output rate and the probabilities of storage levels under the system steady state. These results are the basis of some studies which examine how the storage capacity allocation optimizes the system performance.

Assumptions concerning the stages of systems in existing studies can be roughly summarized into stages which have stationary distributed random cycle times, stages which have constant cycle times with random breakdown times and random repair times, and stages which have stationary distributed random cycle times with random breakdown times and random repair times. The stages in a system can complete work at different

times and depending on the assumptions, will be subject to breakdowns and repairs at random times under the continuous time assumption. Under the discrete time assumption, all jobs begin to be transferred from one stage to the next at the same instant in time and the cycle time is the interval between successive transfers.

Studies employing the analytical approach include (Gershwin et al. 83), which provides analytical results for systems which have up to three stages. (Hillier et al. 91) provides numerical results for systems which have up to five stages. (Gershwin 87) provides approximate results based on a decomposition technique and is theoretically not limited by the number of stages contained in a system. Studies employing the simulation approach include (Marks 87), which gives results for a three stage system. The problem of optimal storage allocation has been studied analytically by (Sheskin 74) and studied using simulation by (Hira et al. 87).

The validity of many of the existing studies depends on the assumption that the system is operating under steady state conditions for most of the time period under consideration (Buxey et al. 73). The steady state of a production system, as defined by (Hatcher 69), is reached when the transition probabilities no longer vary with time. This implies that the average rates of entry into and departure from each state of the system are equal. The states of a production system will be stated in chapter two. In the same chapter, the approaches and methods of some of the existing studies will be summarized. Additional studies can be found in the bibliography by (Perros 85), and at the end of this thesis.

1.3 THE PROPOSED APPROACH

This thesis is an attempt to provide a different approach to the study of production systems with inter-stage storages. The approach is an 'interactive' one which, instead of being based on probabilistic arguments, is established on the instantaneous relationships between the elements in a system. System engineers working in an automobile assembly plant provided the motivation for the current study and it will be presented in chapter three.

The study in (Masso et al. 74) stated that, the existing analytical techniques are too complex and cumbersome and simulation techniques do not give much ground for drawing inferences for the analysis of the production systems. Moreover, because the information provided by the existing studies concentrates on calculating the overall effect of a system based on the assumed characteristics of the stages, insightful information about the operation of the system and the cause and effect relationships are in general, limited. The current approach establishes an independent measurement scheme for the elements of the system. This approach is analogous to that used to analyse electrical circuits. The independent measurements then become the system parameters, comparable to the resistance of an element in an electrical circuit, and are the basis of the formulation of the relationships between the elements and the effect the elements have on the system.

Existing studies have shown that there are inherent complexities with analysing the system. The current study lessens the inherent complexities of analysing the system by devising an abstract representation of the system which will be described in detail in chapter three. The results of this current study show firstly that instantaneous system

variables such as the duration of each blocking, can be obtained exactly through the abstract representation, based on the independent measurement results of the abstract elements. In other words, a system can be represented by an abstract representation with some details about the system omitted in the representation level without losing accuracy. Secondly, this study attempts to answer the questions concerning the origins of production loss and its prevention. The questions posed will be explained in chapter three. Optimally, this study tries to show, by using two practical purpose examples, which are the production control and the storage allocation problems, that the current approach can be treated as a general approach in addition to the existing ones, for the understanding and the answering of questions concerning the system.

1.4 THESIS STRUCTURE

In chapter two, a survey of some of the existing studies will be given.

In chapter three, the problem statement, the assumption, the abstract representation, the element for the abstract representation and the thesis objectives will be given.

In chapter four, the independent measurement of an element will be defined and the calculation of various independent measurements will be derived.

In chapter five, the relationships between the independent measurements and the system parameters, such as, system output, efficiency, duration of blocking and starving will be derived.

In chapter six, the cause and effect relationships of production loss will be quantified among elements. The basic requirement for system monitoring is shown to

be satisfied. The cause and effect relationships for production loss derived in this chapter should be useful for further studies.

Based on the results of chapter six, chapter seven will give two examples. The first example suggests a solution to the production control problem which ensures that production output volume can meet the expected value. The second example suggests a theoretical solution to the storage allocation problem which decides the amount of inter-stage storage under a constraint to maximize production output.

In chapter eight, the experiment of applying the current methodology to systems with parallel configurations will be discussed.

Chapter nine will give a conclusion of the thesis.

1.5 CONCLUSION

In this chapter the traditional assembly line and assembly system with inter-stage storages has been briefly introduced. The argument for a system with inter-stage storages to be balanced has been presented based on the results of a literature survey. Therefore, the system considered in this study is assumed to be balanced.

The difference between the proposed approach and the existing ones has been outlined in this chapter. Where the proposed approach fits into the existing ones has been defined. The proposed approach concentrates on reviewing the instantaneous relationships among the elements in a system and provides insight into the system's operations. Thus, the proposed approach is very different from the existing ones which are mostly based on probabilities. It is hoped that the proposed approach can be accepted as an additional approach which can be used to analyse production systems with inter-

stage storages.

Since this thesis is an attempt to provide a different approach to the study of production systems with inter-stage storages, its objectives therefore are different from those of existing studies. Furthermore, although this thesis is focused on the study of production systems, it is not to be treated as an industrial engineering thesis since the approach taken in this thesis is very different from existing studies in the field of industrial engineering.

CHAPTER 2

SURVEY OF RELEVANT APPROACHES

2.1 INTRODUCTION

In this chapter some examples of existing studies will briefly be reviewed. This chapter presents an overview of the general existing approaches. As mentioned in chapter one, the existing studies are mostly based on probabilities and the approaches can be roughly grouped into the analytical, simulation, and heuristic approaches. Each approach uses a variety of methods and this chapter is by no means an exhaustive illustration of the methods or the approaches used. The examples included in this chapter were chosen because they either represent one of the approaches or because they represent certain interesting points and are useful for illustrating the variety of methods these approaches use.

General assumptions which apply to all of the examples in this chapter (unless otherwise specified) are:

1. There is an inexhaustible supply of workpieces available upstream of the first stage in the system, and an unlimited storage area present downstream of the last stage.

2. Workpieces are not destroyed or rejected at any stage in the system. Partly processed workpieces are not added to the system.

3. Transfer times between stages are negligible

Other major assumptions concerning the stages of the system in the examples are specified individually for each example. At least one example of each approach will be presented as well as two examples about the problem of optimal storage allocation and one example about the production control problem.

2.2 ANALYTICAL APPROACH

The examples given in this section are classified into finite queues in series and systems subject to random breakdowns and repairs. Many methods of the analytical approach are based on the Markov process. The Markov process is briefly introduced below.

2.2.1 Markov Process

According to (Gross & Harris 85), "a discrete-parameter stochastic process $\{ X(t), t = 0, 1, 2, \ldots \}$ or a continuous-parameter process $\{ X(t), t > 0 \}$ is said to be a Markov process if, for any set of n time points $t_1 < t_2 < \ldots < t_n$ in the index set or time range of the process, the conditional distribution of $X(t_n)$, given the values of $X(t_1)$, $X(t_2)$, $X(t_3)$,..., $X(t_{n-1})$, depends only on $X(t_{n-1})$, the immediately preceding value; more precisely, for any real numbers $x_1, x_2, \ldots, x_n$,

$$\Pr\{ X(t_n) \le x_n \mid X(t_1) = x_1, \ldots, X(t_{n-1}) = x_{n-1} \}$$

$$= \Pr\{ X(t_n) \le x_n \mid X(t_{n-1}) = x_{n-1} \}.$$

The process is said to be 'memoryless' because given the 'present' condition of the process, the 'future' is independent of the 'past'.

2.2.2 Finite Queues In Series

A pioneer study of a serial finite queuing model is performed in (Hunt 56). That study initially considers two stages in series. A finite queue of size N is allowed in front of stage 2 while an infinite queue is allowed in front of stage 1. The two stages are not subject to breakdowns and repairs, and cycle times are assumed to be exponentially distributed with rate μ_i for stage i. Arrivals to stage 1 are assumed to follow a Poisson distribution with a mean arrival rate λ, λ being chosen sufficiently large enough so that stage 1 always has a part available on which to work. That study states that "in the general N-stage problem, blocking occurs more frequently in the first stage than in any succeeding stage, and the maximum possible utilization of the first stage is the maximum possible utilization of the entire system". The queuing process is modeled as a continuous time Markov chain. The following N+3 states are defined:

State	Condition of Stage 1	Number of Items in Queue and Stage 2
0	working	0
1	working	1
2	working	2
:	:	:
N+1	working	N+1
N+2	blocked	N+2

Assuming the system is at a steady state, steady state equations are setup and

solved for the probability of having state N+2, which is P(stage 1 block). Then:

$$\rho_{max} = 1\text{-}P(\text{ stage 1 blocked }) = \frac{\mu_2(\mu_1^{N+2} - \mu_2^{N+2})}{(\mu_1^{N+3} - \mu_2^{N+3})} \qquad \mu_1 \neq \mu_2 \qquad 2.1$$

$$= \frac{(N+2)}{(N+3)} \qquad \mu_1 = \mu_2$$

where ρ is a measure of system utilization:

$$\rho = \frac{\text{mean arrival rate}}{\text{mean service rate}} \qquad 2.2$$

The numerical result of ρ_{max} is also given in the study for a three stage system with a queue length of one in front of the second and third stage.

2.2.3 Random Breakdowns and Repairs

The probabilistic production rate of stage i (P_i) is generally expressed as a product of the stand alone mean production rate (μ_i) of the stage and the efficiency (E_i) of the stage. The efficiency of a stage is defined as the probability that the stage is working and not starved or blocked or under repair:

$$P_i = \mu_i \times E_i \qquad 2.3$$

then for a n-stage system, using the 'conservation of flow law':

$$P = P_i \qquad i = 1..n \qquad 2.4$$

where P is the system production rate.

In (Buzacott 72) a study of a system consisting of two stages separated by a size

N storage is performed. The stages in the system are assumed to be identical and independent. The cycle times of each stage are exponentially distributed with parameter μ. The probability of a stage breaking down during a cycle is a constant b, and the repair times of a stage are exponentially distributed with parameter r.

Each stage can be in one of three possible states: the working state (W), the repair state (R), and the idle state (I), when the stage is blocked or starved. A system state is defined as (S_1S_2, X), where S_i is the state of stage i, and X is the inventory level in the storage. The steady state transition equations are set up according to the definition of steady state which means that the probability of departure from a state equals the rate of entering that state. For example, the system leaves state (WW, X) if processing at either station ends. The system can enter the state from either (*i*) WW, X-1 when processing at station 1 ends, and there is no breakdown or (*ii*) WW, X+1 when processing at station 2 ends, and there is no breakdown or (*iii*) RW, X-1 when repair of station 1 ends or (*iv*) WR, X+1 when repair of station 2 ends. This gives the equation:

$$2\mu P(WW, X) = \mu(1-b)P(WW, X-1) + \mu(1-b)P(WW, X+1) + rP(RW, X-1) + rP(WR, X+1) \qquad (0<X<N) \qquad 2.5$$

Likewise, other equations are set up until all of the possible transitions are covered. There are twelve steady state equations for this system. The steady state probabilities of the system can be found by solving all of the equations. The system efficiency then can be found by:

$$P = \mu \left[\sum_{X=0}^{N} \{P(WW, X) + P(RW, X)\} + P(IW, N) \right] \qquad 2.6$$

which is the probability of the second stage being in a working state and not starved or blocked or under repair. The solution of the system production rate is given by:

$$P = \mu \left\{ 1 + \frac{\mu b}{r} + \frac{r^2 - \mu^2 b^2 + 2\mu^2 b}{(N+2)r(r+\mu b) + 2\mu b(\mu - r - \mu b)Jr/(\mu + r)} \right\} \qquad 2.7$$

where

$$J = \frac{(1+u)(1-u^{N+2})}{(1-u)\{1+u^{N+2} - u(1+u^N)a(1-p)/(\mu+r)\}} \qquad 2.8$$

where u is one of the two roots of the quadratic:

$$s^2 - 2vs + 1 \qquad 2.9$$

and

$$v = \frac{r^2 + \mu(\mu+r)(2-b)}{\mu(\mu+r-\mu b)(2-b)} \qquad 2.10$$

However, it is stated in the study that the solution is rather complex and difficult to comprehend. An approximate method to the same problem is also given.

A similar system is analysed in (Gershwin, Berman 81). The stages in that case are assumed to have exponential distributions in service time with the parameter μ, breakdown time with the parameter b, and repair time with the parameter r. The state of stage i is α_i which is defined as 1 if machine i is operational and 0 if machine i is under repair. A system state is defined as (α_1, α_2, X), where X is the inventory level in the storage. P(α_1, α_2, X) is the probability of the system in state (α_1, α_2, X).

The steady state equations are set up according to the definition of the steady state definition. The efficiencies in terms of the steady state probabilities of the two machines are:

$$E_1 = \sum_{X=0}^{N-1} \sum_{\alpha_2=0}^{1} P(1,\alpha_2,X) \qquad 2.11$$

$$E_1 = \sum_{X=0}^{N-1} \sum_{\alpha_1=0}^{1} P(\alpha_1,1,X) \qquad 2.12$$

Under the conservation of flow, the rate of flow out of machine 1 must equal the rate of flow into machine 2. The system production rate is:

$$P = \mu_1 E_1 = \mu_2 E_2 \qquad 2.13$$

It is defined that states with $1 \leq X \leq N\text{-}1$ are internal states and all other states are boundary states. It is shown that solutions for the probabilities of internal states are in a form of:

$$P(n,\alpha_1,\alpha_2) = \sum_{j=1}^{4} c_j X_j^n Y_{1j}^{\alpha_1} Y_{2j}^{\alpha_2} \qquad 2.14$$

where c_j, X_j, Y_{1j}, Y_{2j} $(j=1,2,3,4)$, are parameters to be determined. These parameters are found by substituting the solution form into the steady state equations of the internal states. There are eight boundary states. Solutions for probabilities of four of the boundary states have slightly different forms than those of the internal state solutions, and probabilities for the other four boundary states are shown to be zero. Computational results demonstrate that the model behaves in a manner consistent with intuition.

In (Gershwin, Schick 83), which summarizes two previous studies by the same authors, presents an analytical approach for systems with three stages with storage. Each stage in the system is assumed to have an equal and constant cycle time, a constant probability for breakdown time and repair time. It is claimed that the method benefits from using the structure of the chain to substantially reduce the size of the linear system to be solved. However, as is also stated in the study, a consequence of the method is that there is a loss of sparsity and the problem may become ill-conditioned.

An analytical method of approximate decomposition is performed in (Gershwin 87). It is stated in that study that the exact decomposition method is possible only for systems with infinite inter-stage storages. On the other hand, the Markov chain representation of a n-stage system with n-1 storages has a state space of cardinality:

$$2^n \prod_{i=0}^{n-1} (C_i + 1) \qquad 2.15$$

where C_i is the capacity of storage i. The state space is too large to allow an exact analysis to be performed on many systems. The results sought by the exact analysis can be approximately obtained by the approximate decomposition method in the study by solving 4(n-1) equations.

In the study, each stage in the system is assumed to have an equal and constant cycle time. The probability for breakdown (b_i), and for repair (r_i) are constants for any $stage_i$. Each stage has two possible states, operational or under repair. Repairs and breakdowns occur at the beginning of the cycles, and changes in storage levels take place at the end of the cycles.

The decomposition method uses a set of n-1 two-stage systems L(i) (i=1..n-1),

to represent a n-stage system L with n-1 storages, as is shown in figure 2.1. Each two stage system L(i) is composed of an upstream stage $M_u(i)$, a downstream stage $M_d(i)$, and a storage B(i) which has the same capacity as storage B_i in line L. Both $M_u(i)$ and $M_d(i)$ of L(i) are assumed to have geometric working time distributions, described by parameters $\mu_u(i)$ and $\mu_d(i)$, respectively. Their repair times are assumed to be geometrically distributed with parameters $r_u(i)$ and $r_d(i)$, respectively. $\mu_u(i)$, $\mu_d(i)$, $r_u(i)$ and $r_d(i)$ for each two-stage system L(i) (i=1..n-1), are to be determined. The purpose of decomposition is to have $M_u(i)$ model the part of the original system upstream of B_i, and $M_d(i)$ model the part of the original system downstream from B_i, so that the behaviour of the material flow in the storage of system L(i) approximates that of storage B_i of the original system L.

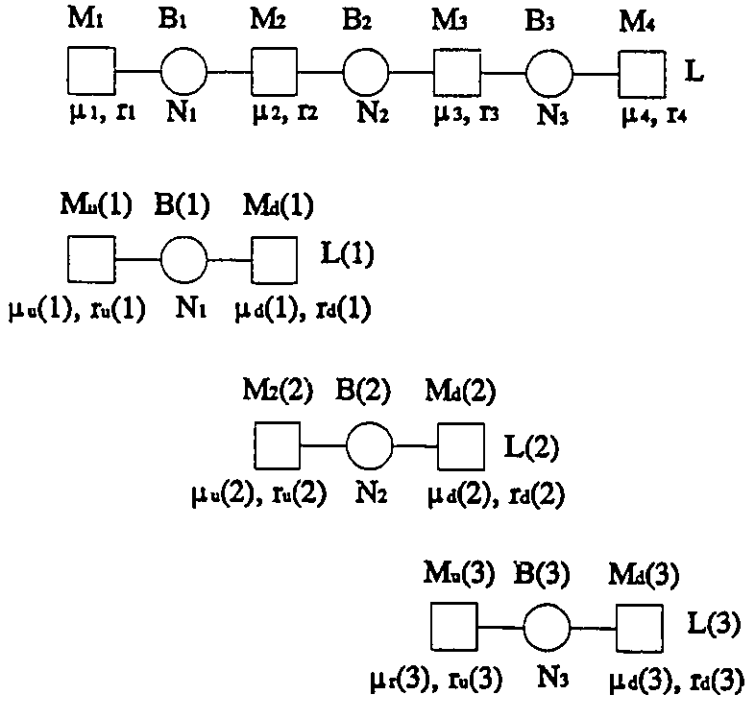


Figure 2.1 A set of 3 two-stage systems for the decomposition of a four-stage system

E(i) is defined as the efficiency of a two-machine line L(i). $p_s(i)$ and $p_b(i)$ are the probabilities of the stage being starved and blocked in L(i), respectively. Based on the exact analysis result of two-machine systems, E(i), $p_s(i)$ and $p_b(i)$ can be calculated as a function of $r_u(i)$, $\mu_u(i)$, $r_d(i)$ and $\mu_d(i)$.

Under the conservation of flow, the study states that:

$$E(i) = E(1) \qquad i = 2..n-1 \qquad 2.16$$

It is assumed that the probability of B_{i-1} being empty and B_i being full is zero:

$$\text{prob}(n_{i-1} = 0 \text{ and } n_i = N_i) \approx 0 \qquad 2.17$$

The following equations have been derived analytically to solve for $r_u(i)$, $\mu_u(i)$, $r_d(i)$ and $\mu_d(i)$:

$$\frac{\mu_d(i-1)}{r_d(i-1)} + \frac{\mu_u(i)}{r_u(i)} = \frac{1}{E(i)} + \frac{1}{e_i} - 2 \qquad i = 2..n-1 \qquad 2.18$$

where e_i is the isolated efficiency of machine i defined in (Buzacott 67) as:

$$e_i = \frac{r_i}{r_i + b_i} \qquad 2.19$$

$$r_u(i) = r_u(i-1)X(i) + r_i(1 - X(i)) \qquad i = 1..n-1 \qquad 2.20$$

where

$$X(i) = \frac{p_s(i-1)r_u(i)}{\mu_u(i)E(i)} \qquad 2.21$$

$$r_d(i-1) = r_d(i)Y(i) + r_i(1 - Y(i)) \qquad i = 1..n-1 \qquad 2.22$$

where

$$Y(i) = \frac{p_b(i)r_d(i-1)}{\mu_d(i-1)E(i-1)} \qquad 2.23$$

and the boundary conditions:

$$\mu_u(1) = \mu_1, \qquad \mu_d(n-1) = \mu_n, \qquad r_u(1) = r_1, \qquad r_d(n-1) = r_n \qquad 2.24$$

The 4(n-1) equations are solved by iteration for the 4(n-1) unknowns, $r_u(i)$, $\mu_u(i)$, $r_d(i)$ and $\mu_d(i)$ (i=1..n-1). The results are shown to be very close to those using exact analysis or simulation. However, it has been reported in some cases that the algorithm failed to converge. Based on the same analysis, (Dallery et al. 88) improved upon the numerical method and reported that there was an improved convergence.

2.3 SIMULATION APPROACH

An alternative approach which can be used to study production systems with inter-stage storages is the simulation approach. The advantages of using simulation include the more flexible choice of the cycle time characteristics of the stages in addition to the normally assumed ones, and flexibility in system size. On the other hand, simulation gives only approximate results on figures such as system efficiency and expected storage levels, because of the finiteness in simulation time. Moreover, the simulation time becomes larger as the system size grows.

The study of (Hira & Pandey 87) focuses on systems with stages that never break down, and have normally distributed cycle times with the coefficient of variation CV. Each inter-stage storage size is set to equal S. Simulations are performed for all combinations of number of stages (n), storage size (S) and coefficient of variation (CV)

as shown below:

$n = 2,3,\ldots,10,12,15,20$

$S = 0,1,\ldots,6,8,\ldots,12,15,20$

$CV = 0.1, 0.2,\ldots,1.0$

Data is collected at the system steady state. System efficiency (E) is derived empirically to be:

$$E_n^s = \frac{1}{S+K+1}\left[S+(K+1)E_2^0-\left(\frac{2-n-2/n}{0.9}\right)(K+1)(E_2^0-E_{20}^0)\right] \qquad 2.25$$

where

$$K \approx -1+2CV+CV^2 \qquad 2.26$$

E_n^S is the efficiency of the system with n stages and individual storage size of capacity S each. E_2^0 and E_{20}^0 are derived from a regression analysis of the simulation data for 2 and 20 stages systems which have zero inter-stage storages, and are shown to be quadratic functions of CV. A similar approach can be found in (Marks 86).

2.4 HEURISTIC APPROACH

An heuristic approach is presented in (Ignall & Silver 77) and is used for estimating the output rate of a two stage system with inter-stage storages. Each stage in the system is assumed to have a fixed production rate μ_i (i=1,2) and is subject to stochastic failures. The probabilities for the breakdown and repair of a stage are assumed to be exponentially distributed with parameters b_i and r_i (i=1,2), respectively. It is assumed that only one machine is repaired at a time.

The output rate of the system which has no storage is defined as R_0, and is defined as R_∞ where there are infinite storages. The study puts forth the argument that if the system has a finite amount of storage capacity (n) between stages then the output rate of the system must lie between R_0 and R_∞ inclusively as n varies between 0 and ∞. Therefore, for any positive n, the system output can be described as:

$$R(n) = R_0 + (R_\infty - R_0)m(n) \qquad 2.27$$

Where m(n) is the weighting factor, with $m(0)=0$, $m(\infty)=1$, m(n) increases as n increases. R_∞, the system output rate with unlimited inter-stage storage is given by:

$$R_\infty = \min\left(\frac{\mu_1}{1+p_1/r_1}, \frac{\mu_2}{1+p_2/r_2}\right) \qquad 2.28$$

R_0, the system output rate without inter-stage storages is:

$$R_0 = \min(\mu_1, \mu_2)\frac{S}{T} \qquad 2.29$$

where T is a sufficiently long operating period, and S is the portion of T that the system is producing. During the S time period, machine i has Sb_i breakdowns and an average of Sb_i/r_i time period is required for a repair. Since it is assumed that only one machine is repaired at a time, the total system down time is then $Sb_1/r_1 + Sb_2/r_2$. System down time also equals T-S. After equating the two expressions for system down time, R_0 becomes:

$$R_0 = \frac{\min(\mu_1, \mu_2)}{1+b_1/r_1+b_2/r_2} \qquad 2.30$$

Based on the result given by (Buzacott 67), m(n) is derived:

$$m(n) = \begin{cases} \dfrac{gn\beta}{gn\beta+1} & \text{if } b_1 = b_2 \\ \max(b_1/b_2, 1)\dfrac{1-(b_1/b_2)^{gn\beta}}{1-(b_1/b_2)^{gn\beta+1}} & \text{if } b_1 \neq b_2 \end{cases} \qquad 2.31$$

where

$$g = 1/2(1+\sigma\beta) \qquad 2.32$$

Defining σ_i to be the standard deviation of repair time distribution for stage i, and if $\sigma_1 \neq \sigma_2$ then:

$$\sigma = \frac{\sqrt{b_1 b_2 (1/r_1 - 1/r_2)^2 + (b_1+b_2)(b_1\sigma_1^2 + b_2\sigma_2^2)}}{(b_1+b_2)} \qquad 2.33$$

$$\beta = \frac{b_1+b_2}{b_1/r_1 + b_2/r_2} \qquad 2.34$$

The results of this method after comparison to those obtained by simulation, are a good estimation. The method is also extended to systems where each stage contains more than one identical machine.

2.5 OPTIMAL STORAGE ALLOCATION

In (Barten 62), a method for the optimal storage allocation problem is presented based on simulations. It is assumed that each stage in the system has normally distributed cycle times, and implicitly assumed that cycle time variations are the only source of randomness. Simulations are run for systems contain two, four, six, and ten stages. For optimal storage allocation, the following diagram has been plotted.

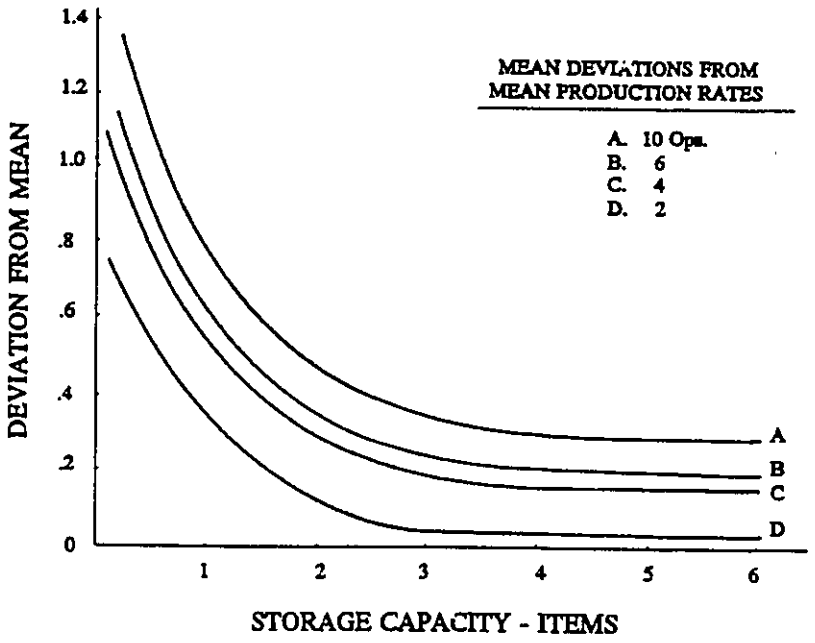


Figure 2.2 Mean deviations from mean production rates at various storage capacities

The vertical axis of figure 2.2 labelled 'deviation from mean' is defined for the output stage of the system as follows:

$$\frac{\text{mean output time - mean stand alone output time}}{\text{standard deviation of stand alone cycle time}} = \frac{\text{mean delay time}}{\text{standard deviation of stand alone cycle time}} \qquad 2.35$$

A general equation is derived to approximate the 'mean delay time' for the range of storage sizes simulated as follows:

$$\text{deviation mean} = c\sigma x^{-0.65} \qquad 2.36$$

Where x is the storage capacity, c is a function of the queue length and σ is the standard deviation of the independent cycle time. For storage allocation, f(t) is defined to be the cost as a function of the mean delay time, and q(x) is the cost of providing a

storage capacity of x. The total cost is:

$$\text{total cost} = q(x) + f(c\sigma x^{-0.65}) \qquad 2.37$$

The objective of optimal storage allocation is to minimize the total cost by using x as the variable.

In the study of (Soyster et al. 79), the optimal storage allocation problem is treated as a maximization problem. Each stage in the system is assumed to have fixed cycle times, and the breakdown of a stage only occurs at the end of a cycle. A stage being operational or broken down is stochastically independent of any other stage. The probability of machine i being operational is denoted by q_i. The steady state probability of obtaining an output per cycle from the system is denoted by $f^*(B)$, where $B = (B_1, B_2, \ldots, B_{n-1})$ is the storage allocation assignment.

The objective of the study is to maximize $f^*(B)$, such that $(AB \leq b)$ and $(B_i \geq 1)$ $(i=1..n-1)$. Where A is an arbitrary (m x n-1) matrix, it is the constraint relating to the cost of providing storages. From a previous study by the author, the form of f^* is only known for the case of a two stage system. Based on f^* for a two stage system, $f_i(B_i)$ is defined for a two machine sub-system composed of stage i, storage capacity B_i, and stage i+1:

$$f_i(B_i) = \begin{cases} \dfrac{q_i q_{i+1}(1-\rho_i^{B_i})}{q_{i+1}-q_i\rho_i^{B_i}} & \text{if } q_i \neq q_{i+1} \\ \dfrac{qB_i}{B_i+1-q} & \text{if } q_i = q_{i+1} = q \end{cases} \qquad 2.38$$

where

$$\rho_i = \frac{q_i(1-q_{i+1})}{q_{i+1}(1-q_i)} \qquad 2.39$$

It is shown that f^* is bounded as:

$$\prod_{i=1}^{N} q_i \le f^*(B) \le \min_{1 \le i \le N} q_i \qquad 2.40$$

Based on the boundaries of f^*, the three functions, f, g and h are used as the surrogate functions of f^*, since f^* is not known:

$$f(B) = \min_{1 \le i \le n-1} f_i(B_i) \qquad 2.41$$

$$g(B) = \prod_{i=1}^{n-1} f_i(B_i)^{[\min(q_i, q_i+1)]^{-1}} \qquad 2.42$$

$$h(B) = K \prod_{i=1}^{n-1} f_i(B_i) \qquad 2.43$$

where K is a constant. These three functions are used to substitute f^* for the maximization problem. Systems with 4, 6 and 8 stages are then studied. After the results are compared with other existing results, it can be seen that neither f, g or h necessarily provide the theoretical optimum.

The study by (Hira & Pandey 87) mentioned earlier in this chapter also gives an empirical formula for finding the optimal storage allocation based on simulation results. The constraints are a set of assumed costs, and the objective is to maximize the gain which is defined as the difference between the additional revenue of adding one unit of inter-stage storage and the total cost.

2.6 PRODUCTION CONTROL

Although the study concerning production control does not fall into the categories which have been used for the survey of the existing studies, it is mentioned here because the problem of production control is examined by one of the application examples of this thesis. The existing study used as an example here was published when the draft of this thesis was in the process of being completed. Therefore, the categories and some of the comments for the existing studies may not necessarily apply to this example.

The objective of production control has been defined in (Van Ryzin et al. 93) as the operation of the system so that it meets the demand without overproducing, underproducing or carrying large quantities of material in the storage. In that study, the problem was investigated using a two machine model which has an infinite storage in between the machines. The machines are subject to stochastic failures and repairs. The state of machine i ($i=1,2$) is denoted by $\alpha_i(t)$, where $\alpha_i(t)=1$ if machine i is operational and $\alpha_i(t)=0$ otherwise. The production rate of machine i is denoted by $u_i(t)$ and is constrained by the maximum production rate U_i. The demand rate is denoted by d. The production surplus at machine i is defined as follows:

$$x_i(t) = \int_0^t [\, u_i(\sigma) - d \,]\, d\sigma, \quad i=1,2 \qquad 2.44$$

The optimal control problem was formulated as follows:

$$\dot{x}_1(t) = u_1(t) - d \qquad 2.45$$

$$\dot{x}_2(t) = u_2(t) - d \qquad 2.46$$

$$x_2(t) \le x_1(t) \qquad 2.47$$

$$0 \le u_1(t) \le \alpha_1(t)U_1 \qquad 2.48$$

$$0 \le u_2(t) \le \alpha_2(t)U_2 \qquad 2.49$$

$\Omega(t)$ was defined as the set of $u=\{u_1,u_2\}$ which satisfy equations 2.48 and 2.49. The objective was to minimize the cost function defined as follows:

$$\min_{u \in \Omega(t)} E \int_0^{\infty} e^{-\beta t} g(x_1(t), x_2(t))dt \qquad 2.50$$

A numerical solution for a discrete-time version of the problem has been obtained with d chosen to be 1 per unit time and $e^{-\beta}$ chosen to be 0.9. The results were plotted as optimal control region diagrams. The system state was represented by $\alpha_i(t)$ and $x_i(t)$, $(i=1..2)$. An optimal control region diagram is plotted for each combination of $\alpha_1(t)$ and $\alpha_2(t)$ while $x_1(t)$ and $x_2(t)$ made up the two axes of the diagram. With the case of $\alpha_1(t)=\alpha_2(t)=1$, there are four control regions in the diagram. Depending on which region the current system state belongs to, the optimal control action is one of the four combinations of having machine$_i$ $(i=1..2)$ produce at its maximum rate or produce nothing at all. An approximate control based on the characteristics observed in these results was also developed.

2.7 CONCLUSION

In this chapter a brief overview of the existing approaches to the study of production systems with inter-stage storages has been presented. Existing approaches have been roughly summarized into the analytical, simulation and heuristic approaches. At least one example has been presented for each of the approaches.

The existing approaches have concentrated on the calculations of the overall probabilistic results of a system based on the assumed characteristics of the stages. Therefore, instantaneous cause and effect relationships among the stages and the system output are not obvious. Moreover, it has been stated that the existing analytical techniques are too complex and cumbersome, and it is believed that simulation techniques do not provide grounds for drawing inferences (Masso et al. 74). It is the author's theory that the instantaneous relationships are the key to obtaining further insights into the operations of a system. Therefore, the proposed approach contained in this thesis provides a different point of view about the system by deriving the instantaneous relationships of a system through the simplification provided by the use of an abstract representation.

The proposed approach emphasizes the measuring of the local effects of a stage instead of assuming the cycle time distributions, as has been done in many studies. The usual assumptions about service times, failure-free operation times and repair times of the stages in a system are shown to be unrealistic, as was pointed out in (Buxey et al. 73). That study states that the exponential distribution is not particularly representative of operator variability but does provide for mathematical manipulation. It is further shown in (Vasil'ev et al. 89) that an exponential distribution failure-free operation time

is a special case which is true only when element lifetimes are also distributed exponentially. Other discrepancies between the assumptions about stage cycle times and reality can be found in (Buzacott, Hanifin 78).

CHAPTER 3

OVERVIEW OF THE PROPOSED METHODOLOGY

3.1 INTRODUCTION

This chapter outlines the problem considered by the current study and states the assumptions about the system considered by the study. A methodology will be proposed based on an abstract representation of the system. The purpose and reason for using an abstract representation is explained. The objectives of the thesis are stated.

3.2 THE PROBLEM

The proposed approach for studying production lines with finite inter-stage storages is a novel one and one which is motivated by the need for on-line system monitoring. The on-line monitoring problem of assembly lines with inter-stage storages is an important one, since the importance of the assembly line and the use of inter-stage storages has been justified as early as in (Buzacott 67), (Anderson et al. 69). The purpose of on-line monitoring is to observe whether or not the production system can carry out certain design characteristics, or to explain the differences otherwise, during production. Many studies have been devoted to the understanding of design stage problems but the problem of on-line monitoring has attracted little attention.

One question concerning system monitoring, for example, is why the output

volume of a system does not meet its expected value for a certain production period. The expected output volume of a system can be designed analytically or by using simulation, assuming that there is a certain cycle time distribution for each stage in the system. The expected output volume and the efficiency of the system should then be bounded by the designed value after the system has reached a steady state. However, if the cycle times of some stages in the system do not follow the assumed cycle time distribution (as mentioned in chapter two), or the system does not reach a steady state, then the system's efficiency and the expected output volume could be different than is expected.

From a practical point of view, a system which fails to meet its output volume can be costly since overtime or extra production shifts may be required. On the other hand, from the management's point of view, it is important to know what prevents the system from meeting its target, where the problem lies and what changes can be made to the run-time operations to improve the situation. Answers to the above questions demand quantitative associations between the causes and effects of the problem. Any such relationships should be produced based on real time values concerning the system, since problems due to run-time operations can be different under different circumstances, and are very unlikely to be repeated. Real time results also enable the people managing the system to know how to avoid production loss as production goes on. Therefore, the proposed methodology uses an interactive approach.

The problem of on-line system monitoring is not a trivial one. The complications of system design caused by, among others, variations in processing times, finite storage space, inter-dependence of stages applies equally to the on-line system monitoring

problem. The complexity of system monitoring is due to the existence of inter-stage storages which create a 'cushioning' effect on the causes of production loss. It is difficult to observe the system production loss when it occurs inside the system and its formation would be very complex once it effects the system output. On the other hand, system monitoring is not just an observation of the production rate of any stage. It is to determine the effect the performance of any stage has on the system based on the history, the current state of the system and the inter-relationships among stages.

After the instantaneous relationships between the causes and effects of production loss are derived to address the problem of on-line system monitoring, a natural extension of this would be to apply these relationships to address the problems concerning system performance improvements. Therefore, the second and third problem this thesis will address are system production control and storage allocation.

The goal of production control as defined in (Ryzin et al. 93) is to operate the system so that it meets the demand without overproducing, underproducing or carrying large quantities of material in the storage. Large finished goods inventories which are expensive to maintain are the result of overproduction. Unfulfilled orders, which can be even more costly if they result in order cancellations or loss of goodwill are the result of underproduction. In (Ryzin et al. 93), the model consisted of two machines which had an infinite storage between them. The machines were subject to stochastic failures and repairs and the objectives were to control the system so that one job would be produced in one unit time and to minimize the inventory in the infinite storage, as summarized in chapter 2. In chapter seven of this thesis, an algorithm will be derived to address the problem of production control. This algorithm is based on the instantaneous relationships

between the causes and effects of production loss and will be designed to theoretically accommodate systems of any size with arbitrary cycle time behaviours. The storage capacities of the system are finite and can have any value. Therefore, the minimization of inventory being carried in an infinite storage is not part of the objective of the production control algorithm in this thesis. This production control algorithm will calculate the maximum allowable cycle time for each stage for each cycle, so that if these maximum allowable cycle times are met by the stages, the goal of production control can be achieved. Furthermore, the maximum allowable cycle times calculated by the production control algorithm in this thesis will be bounded below so that the maximum cycle time values should fall within an acceptable range.

The goal of storage allocation in a system is to find the amount of storage space between the stages which yields the maximum system efficiency for a certain type of cycle time distribution while satisfying a set of pre-defined constraints. In this thesis, a preliminary study will be undertaken for this problem with the constraint being that the total available storage space is fixed. This preliminary study tries to show that the approximated long term relationships between the causes and effects of production loss can be helpful in providing insights which help solve the problem of storage allocation.

3.3 ASSUMPTIONS

Under the proposed approach, the assumptions applied to the system are:

1. the production line is 'notionally' balanced, meaning that all serial stages in the system should have equal mean processing times.

2. transfer times between any two stages are constants.

3. jobs entering the system remain in the same sequence until they leave, no job is destroyed or discarded.

4. there is an infinite supply of jobs before the first stage, and an infinite amount of space after the last stage of the system.

The first assumption is based on the fact that inter-stage storages can only be effective if the system is 'notionally' balanced. This assumption is realistic and does not impose any unnecessary restrictions on the system.

The second assumption is an idealistic one, since transfer times between stages are likely to vary due to the imperfection of the transportation system. However, the assumption is made to simplify the analysis of the system. Nevertheless, this assumption is relatively realistic compared to those of many existing studies which assumed that the transfer time, whether explicitly or implicitly, is zero.

The third assumption is also an idealistic interpretation of the system, since situations excluded by the assumption, although unusual, may happen. However, the chance that this situation may realistically happen should be small since it could directly result in waste or loss of production. The purpose of the second and third assumptions is to make the cycle times of stages in the system the only variables.

After the methodology has been completely developed for the system, it can be seen that the fourth assumption is not necessary. However, this assumption is necessary for the definition of a system, since the system considered in the study should be independent and isolated from other systems.

An assumption about cycle time distribution is not necessary, since the effect of cycle time variations is measured on a real time basis.

3.4 THE ABSTRACTION

It is well known that production lines with finite inter-stage storages are complicated systems. The complications are due to, among others, processing time variations, finite storage space and inter-dependence of stages. Existing exact analytical methods for calculating system steady state efficiency are limited to systems with three stages because of the complicated nature of the system.

The purpose for introducing an abstract representation is to simplify the system but still retain enough information to represent the system for problem solving. For example, under the current assumptions, a job in the system sitting at a storage will always move forward to its downstream location (either a storage or stage) as soon as one becomes available. This information is essential to the system but does not necessarily help solve the problem. With this methodology, a stage and some of its surrounding storage spaces are grouped together to form an abstract element which can be used to represent the system. Under the abstract representation, only the elements are visible when analysing the system. Thus the elements hide the detailed information inside of them. It will be shown that the simplification inspires an understanding of the various relationships between the elements of a system, an understanding which may otherwise be obscured by complicated details. Figure 3.1 shows a four stage system with inter-stage storages, and its abstract representation.

$Element_1$ to $Element_4$ are the abstract elements. Each element contains exactly one stage and some of its neighbouring storages. Each stage must belong to one and only one element. Storage units between any two stages are divided between the two elements which include the two stages. The total storage space between any two stages

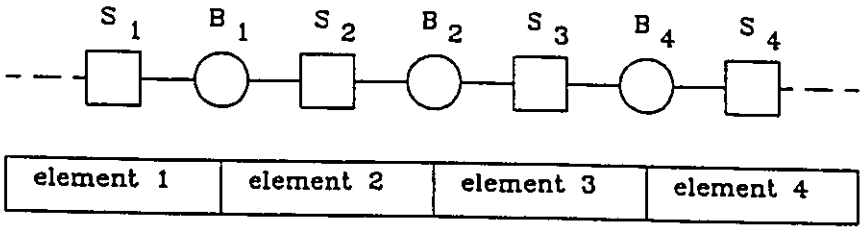


Figure 3.1 A four stage system and its abstract representation

is treated as a continuous space. Thus, the total storage space can be arbitrarily divided up between the two elements. All of the storage spaces must be uniquely included in some element and no element can contain any space that is included by another element. The whole system is then represented only by elements.

Without the abstraction, the stand alone efficiency measurement of a stage is given in (Buzacott 67) by assuming exponential cycle times, failure times and repair times of each stage. However, with the abstract representation, a new way of measuring the stand alone efficiency of each stage, more precisely, each element is needed because each stage is now embedded in an element. The measurement of an element should include the characteristic of the stage and the effect of the storage belonging to the element. With the abstract representation, the system can be treated as a n-element system where n equals the number of stages in the original system.

It is proposed that the independent measurements which are obtained for each of the abstract elements should be used for analysing the system. According to the assumptions, the independent measurement of an element has the stage cycle time as it's only variable.

3.5 THESIS OBJECTIVES

The approach using the proposed abstraction and the independent measurement of each element should theoretically be beneficial since a simplified system is the result. An interactive method is proposed for analysing the system, since on-line system monitoring is the motivation. The purpose of the interactive method is to give results based on the current status of the system and before the status changes.

The proposed methodology first defines the independent measurement for each element. The relationships between the independent measurements and system parameters such as system output, efficiency, duration of blocking and starving in the system are then derived. The system is then completely represented by the abstraction. The cause and effect relationships of production loss in the system can then be quantified among the elements of the abstract representation.

The first objective of the study is to derive the exact readings of the independent measurement pertaining to each element in the ideal world and to find the exact relationships between the independent measurement and the real world system parameters.

The second objective is to show that the derived information can be used to answer the on-line system monitoring problem. This requires the cause and effect relationships of production loss among the elements in a system to be quantified.

The third objective is to show that the derived information can be used to obtain a viable solution for the production control problem and hence providing a validation for the representation.

The fourth objective is to show that this methodology can be applied to gain a

new and different insights into the problem of storage allocation.

3.6 CONCLUSION

In this chapter, the problem statement of the study has been presented. The assumptions about the system being studied have been stated. An abstract representation, which provides a simpler system for later analysis, has been proposed. The system will be analysed using an interactive approach based on the proposed methodology. An overview of the approach taken by the proposed methodology has been presented. The objectives of the study have been stated.

The proposed methodology for representation and on-line analysis of a system is illustrated in figure 3.2. The figure consists of two blocks outlined by the two dashed-line boxes. The lower box contains the system to be analysed and the upper box contains the proposed procedures. The following notes give more details about each layer in the figure:

layer 0: The production system to be analysed. This can be an emulated system such as the one in this thesis for carrying out the development of the methodology or it can be a real system when the proposed methodology is ready to be applied in a real production plant. Each cycle time of each stage is sent to the first layer procedures.

layer 1: The 'abstract representation' for the system as explained in this chapter. It receives a cycle time at the end of each cycle of each stage.

layer 2: The 'independent measurement' or the 'ideal world measurement'. This part of the methodology will be presented in chapter four of this thesis and is part of the first objective of this thesis.

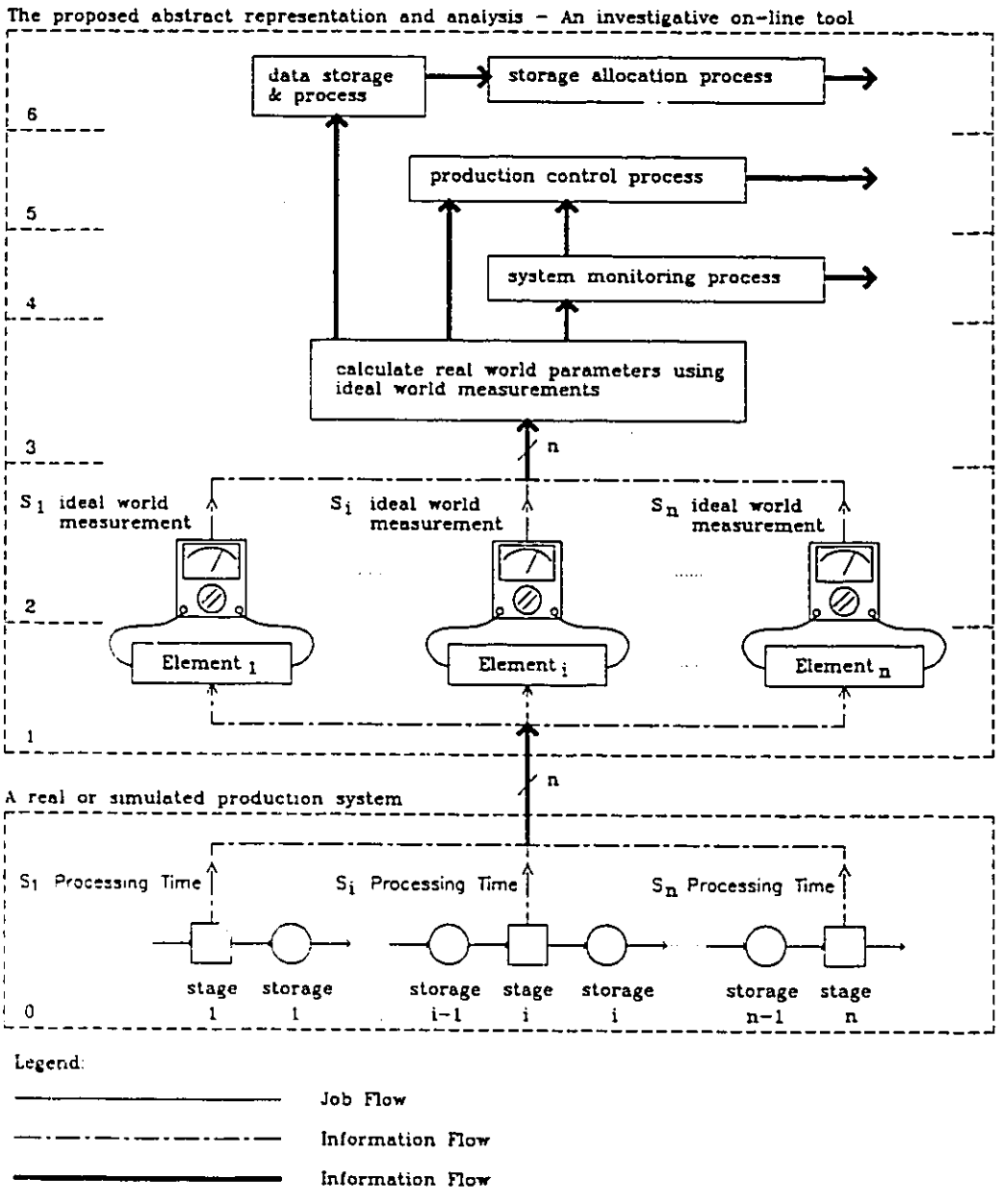
layer 3: Calculation of the real world parameters using the 'ideal world measurement'. The original production system is reconstructed in this layer based on the 'ideal world measurement' and the abstract representation of the methodology is then completed. This results in a simplified system which contains only the abstract elements and the simplified system mimics the original production system exactly in the areas which are vital to the later analysis. This layer will be presented in chapter five and is part of the first objective of this thesis.

layer 4: The 'system monitoring process'. This layer uses the information from layer three and its output contains information pin-pointing the origins of production loss which is experienced at the output of the production system. This information is helpful in on-line system monitoring. This layer will be developed in chapter six and is the second objective of this thesis.

layer 5: The 'production control process'. This layer uses the information from layers three and four. Its output suggests a maximum cycle time for each stage at the beginning of each cycle. This suggested maximum cycle time

for a stage can be displayed and it can possibly increase but never decrease during the same cycle. If each stage abides by the maximum cycle time for every cycle, the system should have a production time close to the expected time for a fixed amount of production. This information can be helpful to production control. This layer will be developed in chapter seven and is the third objective of this thesis.

layer 6: This layer contains two processes: 'data storage and process' and the 'storage allocation process'. 'Data storage and process' will not be discussed in this thesis. Its presence in the diagram is to clarify that the 'storage allocation process' is an off-line process. The 'storage allocation process' uses the stored and processed long-term information gathered from layer 3. The output information of this layer provides insights which can be used to verify the current storage allocation or to improve the system design. The 'storage allocation process' will be discussed in chapter seven and is the first step towards satisfying the fourth objective of this thesis.



e 3.2 Block diagram for the proposed abstract representation and analysis

CHAPTER 4

THE INDEPENDENT MEASUREMENT

4.1 INTRODUCTION

In this chapter, the independent measurement of an abstract element is described in detail. The performance of an element in a system is affected by all of the other elements in the same system. The purpose of the independent measurement is to measure the performance of an element independently, isolating the effects of other elements. The independent measurement must be consistent when applied to different elements and also be consistent with time.

4.1.1 The Ideal World Definition

One of the stated assumptions is that the system is 'notionally' balanced (Buxey et al. 72), which means that the mean cycle times of all of the stages are equal in a sequential configuration. Based on this assumption and the requirement that the independent measurement be consistent with time and with different elements, the following measurement scheme has been derived. It is called the 'ideal world' measurement and it is shown in figure 4.1 for an element.

An element is said to be in an 'ideal world' if it is connected to an input and output where the respective inter-arrival and inter-departure times are equal to the mean

processing time of the stage inside the element. The 'ideal world input' and the 'ideal

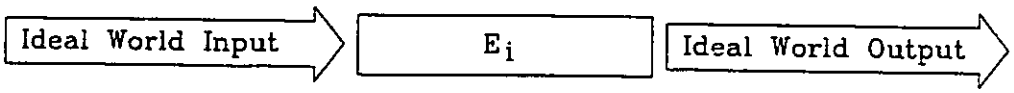


Figure 4.1 An element being measured by an "ideal world"

world output' shown in figure 4.1 using the large arrows are such devices. The 'ideal world input' sends a job into the element at a fixed interval equal to the mean processing time of the stage. There is an infinite supply of jobs inside the 'ideal world input'. The 'ideal world output' takes in a job from the element at a fixed interval equal to the mean processing time of the stage. There is an infinite amount of space inside the 'ideal world output'.

Figure 4.2 shows the contents inside an element which is under the ideal world measurement. Inside the element, the storage connecting the 'ideal world input' and the stage is the input storage of the element. Likewise, the storage connecting the 'ideal world output' and the stage is the output storage of the element.

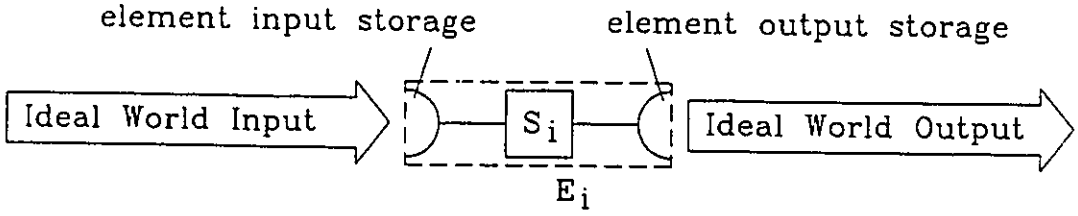


Figure 4.2 The content of an element and the "ideal world" measurement

The storage space of an element is specified in real numbers, which allows for the division of the storage space between any two stages to be arbitrarily chosen when

setting up the elements. The initial contents of the storages and the transfer time between any two stages are divided likewise between the two elements containing the two stages. The independent measurement of an element can then be obtained with this setup.

4.1.2 The Ideal World Measurement

Under the abstract representation, only abstract elements exist in a system. Stages and storages exist only inside the elements. Therefore, the ideal world measurement is applied to an element and not to a stage. The information given by the independent measurement takes into account the effects of the storages inside the element and measures the 'buffered' effect of the stage of the element which is surrounded by the storages.

The result of the ideal world measurement should be determined analytically. On the other hand, if the storage space and contents are integer units, an emulation package for queuing models can be used to find these values for an element. However, it would be computationally more demanding than if done analytically, since it requires a separate and concurrent emulation running for each of all of the elements. Furthermore, it would not allow non-integer values to be used for storage space and contents, which imposes unnecessary restrictions on the abstract representation.

The ideal world measurement results and the ideal world status update of an element are determined by the processing time of the stage in the element. Parameters of an element such as storage size and transfer time are fixed once the abstract representation of the system has been set up.

The parameters of an element under the ideal world measurement do not

necessarily correspond to those of the actual system, except when it is initially set up, since the ideal world is a unique environment isolated from the real environment. For example, input wait can describe the amount of time a stage in an element waited for a job from the 'ideal world input' in the ideal world, and the amount of time a stage waited for a job from its upstream stage in the real world. The two parameters from the two worlds do not necessarily equal each other. It is therefore important to distinguish between the ideal world parameters from those in the real world. The scope of this chapter focuses only on the elements in the ideal world. Therefore, any variable names used in this chapter describe only the element in the ideal world and are not to be confused with the corresponding ones in the real world. A comprehensive convention used to distinguish the parameters of the ideal world from those of the real world, and also one element from another, will be described in the next chapter when the real world parameters are discussed.

4.2 IDEAL WORLD MEASUREMENT DERIVATION

The ideal world measurement of an element is taken after every processing cycle is finished. The storage contents and the job currently being processed are treated as fluid flows in order to analytically derive the results. This allows the storage space and contents to be treated as continuous values. The basic idea is to update the input and output storage contents of an element based on the current cycle time and the constant processing rates of the 'ideal world input' and the 'ideal world output'. The current value of the storage contents changes as functions of the processing time of an element. For the output storage of an element (ob), it can be updated as:

$$ob = `ob - \frac{ct}{mct} + 1 \qquad 4.1$$

The (`) appearing in the upper left corner of a parameter represents the current value of that parameter before updating. (ct) is the processing cycle time of the stage inside an element and (mct) is the mean processing cycle time of the stage which is also equal to the ideal world processing time.

proof: During ct, the 'ideal world output' is consuming the output storage at a rate of 1/mct, and one job is added to the output storage when the stage finishes producing the current job. [*EndofProof*]

Similarly, the input storage content of an element (ib) is updated as:

$$ib = `ib + \frac{ct}{mct} - 1 \qquad 4.2$$

proof: During ct, the 'ideal world input' is producing and adding to the input storage at a rate of 1/mct, and one job is taken from the input storage when the stage finishes producing the current job. [*EndOfProof*]

By definition, the job currently being processed by the stage inside an element is included in (ib) and the job currently being processed by the 'ideal world output' is included in (ob). Therefore, an element has (ob-1) output storage content, and (ib-1) input storage content. The total output storage space of an element is (ob_limit). The total input storage space of an element including the space occupied by the stage is (ib_limit). Therefore, the total input storage space of the stage of an element is (ib_limit-1).

The above updates of (ob) and (ib) of an element are only valid if there is no

delay or wait for the element in the ideal world. However, considerations are required for the consequences of each occurrence of delay or wait in the ideal world update. These considerations will be discussed in detail in the following sections so that the ideal world measurement derivation can be complete. For the above two equations and the subsequent derivations, all parameters of an element are updated from their current values recorded at the start cycle time of the current job of the stage to the values at the start cycle time of the next processing cycle of the stage.

4.2.1 Ideal World Output Delay

An ideal world output delay (output_delay) occurs when the 'ideal world output' of an element has to wait for a job because the output storage of the element is empty. The duration of the ideal world output delay starts from the time the 'ideal world output' demands a job to the time it receives a job.

If a non-zero output transfer time of an element is allowed, an ideal world output delay will happen if the difference between the time the stage finishes a job and the time the 'ideal world output' demands a job is less than the output transfer time, when the output storage is empty, since a job finished by a stage would not reach the 'ideal world output' immediately. The ideal world output delay is shown in figure 4.3 by the timing diagrams of the stage inside the element, and the 'ideal world output' of the element.

In the 'ideal world output' process timing diagram, 'x' is equal to the length of the output transfer time plus the time the stage finishes a job. The time that the ideal world output storage is empty and the 'ideal world output' demands a job is called (ob empty time). There will be an ideal world output delay if (ob empty time) happens

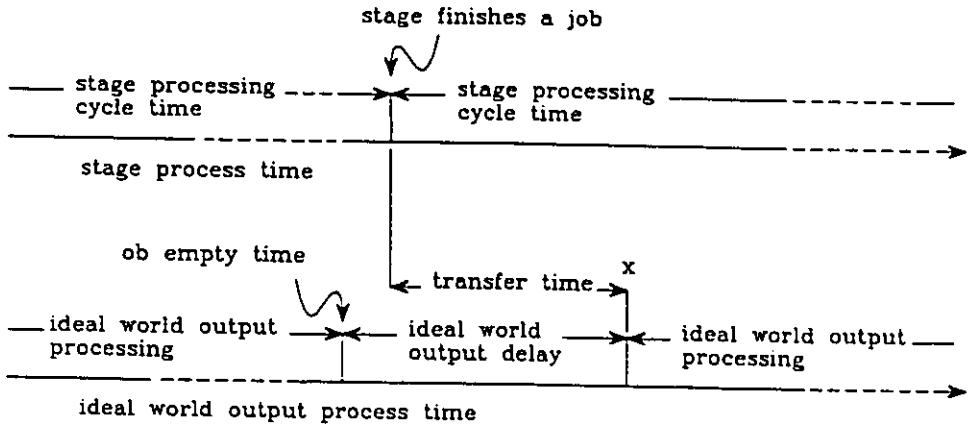


Figure 4.3 Illustration of ideal world output delay

earlier than x. Therefore, (ob empty time) needs to be determined in order to calculate the ideal world output delay. Defining the current start cycle time as the reference point of (ob empty time), and given the value of ('ob) at the current start cycle time of the stage, then:

$$\begin{aligned} &\text{ob_empty_time} = \text{`ob} \times \text{mct} + \theta \\ &\theta = \min(\text{previous_output_delay}, \text{output_transfer_time}) \end{aligned} \qquad 4.3$$

where previous_output_delay is the output delay of the previous processing cycle.

proof: Assuming θ is zero, (ob) is defined as including the current job being processed by the 'ideal world output', and the 'ideal world output' is processing at a rate of 1/mct.

Assuming the ideal world output delay of the previous processing cycle is non-zero ($\theta > 0$), then the previous cycle, after the stage finished a job and ended the previous output delay, started the next job immediately (ignoring for now, the possibility that the input storage could be empty). Using figure 4.3 also for the (output_delay) of

the previous cycle, the 'ideal world output' has to wait until the time equals 'x' to start the next job, since the output transfer time is non-zero. θ is the amount of time the stage starts ahead of the 'ideal world output' on a new job after an output delay. This amount of time is subtracted from the current output delay since during this time, the 'ideal world output' is idle. [*EndOfProof*]

The value of θ equals the output transfer time in the situation illustrated by figure 4.3. An ideal world output delay will happen if (ct + output_transfer_time > ob_empty_time), since both the processing cycle time and the (ob empty time) refer to the current start cycle time of the stage. Defining *r*output_delay (raw output delay) as follows:

$$r\text{output_delay} = \text{ct} + \text{output_transfer_time} - \text{ob_empty_time} \qquad 4.4$$

Defining:

$$\text{pos}(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \qquad 4.5$$

The ideal world output delay is defined as:

$$\text{output_delay} = \text{pos}(\, r\text{output_delay}\,) \qquad 4.6$$

With the ideal world output delay defined, the ob update becomes:

$$\text{ob} = \begin{cases} 1 & \text{if output_delay} > 0 \\ \text{`ob} - \dfrac{\text{ct} - \theta}{\text{mct}} + 1 & \text{otherwise} \end{cases} \qquad 4.7$$

proof: Assuming output_delay > 0, the job currently being processed by the 'ideal world output' is included in (ob), then (ob) must be one after an ideal world output delay

has occurred. Assuming otherwise, the 'ideal world output' is idle and not consuming from the output storage during the time period of θ. [*EndOfProof*]

At the end of each processing cycle of a stage, the output storage of the element will always have content greater than or equal to one. If the output storage content is less than one, an output delay will happen. This can be verified by substituting the expression (ob empty time) into (ct) in the general (ob) update equation. The actual number of jobs belonging to the output storage of the element is one less than the value of (ob), according to it's definition.

4.2.2 Ideal World Output Wait

An ideal world output wait (output_wait) occurs when the stage of an element has to wait for a space because the output storage of the element is full. The duration of the ideal world output wait starts from the time the stage finishes a job to the time the space in the output storage becomes large enough to hold a job. The output transfer time has no effect on the ideal world output wait.

(ob fall time) is defined to be, with reference to the current start cycle time of the stage, the instant marking the transition of the output storage of an element when it changes from full to having just enough space for one job. Ignoring for now the fact that the input storage can be empty:

$$\text{ob_fall_time} = (\text{`ob} - \text{ob_limit}) \times \text{mct} + \theta \qquad 4.8$$

where (ob_limit) is the size of the output storage, which is fixed when an element is defined.

proof: The output storage content of an element has a maximum of one plus the output storage limit, since it is defined to include the job being currently processed by the 'ideal world output'. If (ob) is greater than (ob_limit), the output storage is full. This situation will last until (ob) drops to equal (ob_limit), given that the 'ideal world output' is consuming at a rate of 1/mct. When (ob) equals (ob_limit), the 'ideal world output' has just made enough space for one job in the output storage of the element.

If the ideal world output delay for the previous processing cycle is positive, the 'ideal world output' is idle during the first θ time units, since all of the variables are referring to the start cycle time of the stage in an element. [*EndOfProof*]

If the value of (ob fall time) is negative, a transition of the output storage from full to having one space is not possible for the current cycle. An ideal world output wait will happen if (ob_fall_time > ct). *r*output_wait (raw output wait) is defined as:

$$r\text{output_wait} = \text{ob_fall_time} - \text{ct} \qquad 4.9$$

The ideal world output wait is:

$$\text{output_wait} = \text{pos}(\ r\text{output_wait}\ ) \qquad 4.10$$

proof: If the stage of an element finishes its current job earlier than the transition described by (ob fall time), then it has to wait until (ob fall time) for a space in the output storage. [*EndOfProof*]

The update of ob now becomes:

$$ob = \begin{cases} 1 & \text{if output_delay} > 0 \\ ob_limit + 1 & \text{if output_wait} > 0 \\ `ob - \dfrac{ct - \theta}{mct} + 1 & \text{otherwise} \end{cases} \qquad 4.11$$

proof: Assuming output_wait > 0, at the end of an output wait, the output storage has exactly one space (ob=ob_limit), and it becomes full (ob=ob_limit+1) again after the pending job from the stage is transferred. [*EndOfProof*]

4.2.3 Ideal World Input Delay

An ideal world input delay (input_delay) occurs when the 'ideal world input' of an element has to wait for a space because the input storage of the element is full. The duration of the ideal world input delay starts from the time the 'ideal world input' is blocked until the time a space becomes available in the input storage of the element. The input transfer time has no effect on the ideal world input delay.

(ib full time) is defined to be, with reference to the current start cycle time of the stage in an element, the instant when the input storage of an element is full and the 'input ideal world' becomes ready to transfer a job. Given the value of ('ib) at the current start cycle time, then:

$$ib_full_time = (ib_limit - `ib + 1) \times mct \qquad 4.12$$

where (ib_limit) is the fixed size of the ideal world input storage of an element.

proof: The input storage becomes full when (ib) equals (ib_limit). Then after exactly one (mct), the 'ideal world input' is ready to transfer another job and becomes blocked, since the inter-arrival time of jobs from the 'ideal world input' is mct. [*EndOfProof*]

An ideal world input delay will happen if (ct > ib_full_time). *r*input_delay (row input delay) is defined as:

$$r\text{input_delay} = ct - ib_full_time \qquad 4.13$$

The ideal world input delay will be defined in the section 'ideal world output wait and input storage content'. The update of ib now becomes:

$$ib = \begin{cases} ib_limit & \text{if input_delay} > 0 \\ `ib + \dfrac{ct}{mct} - 1 & \text{otherwise} \end{cases} \qquad 4.14$$

proof: Assuming input_delay > 0, at the end of an input delay, a job is transferred from the 'ideal world input' to the input storage of an element. At the same time, the stage of the element also takes in one job from the input storage, leaving (ib) to equal (ib_limit). [*EndOfProof*]

The fact that after an input delay, the input storage content of an element equals the input storage size, can be verified by substituting (ib full time) for (ct) using the general (ib) update equation.

4.2.4 Ideal World Input Wait

An ideal world input wait (input_wait) occurs when the stage of an element has to wait for a job because the input storage of the element is empty. Ignoring the possibility that the output storage may be full for now, the duration of an ideal world input wait starts from the time the stage of an element finishes the current job and demands a new job, to the time a new job arrives at the stage.

A non-zero input transfer time affects the ideal world input wait in a way similar

to the ideal world output delay. When the input storage is empty, the ideal world input wait is shown in figure 4.4 using the timing diagrams of the stage of an element and the 'ideal world input' of the element.

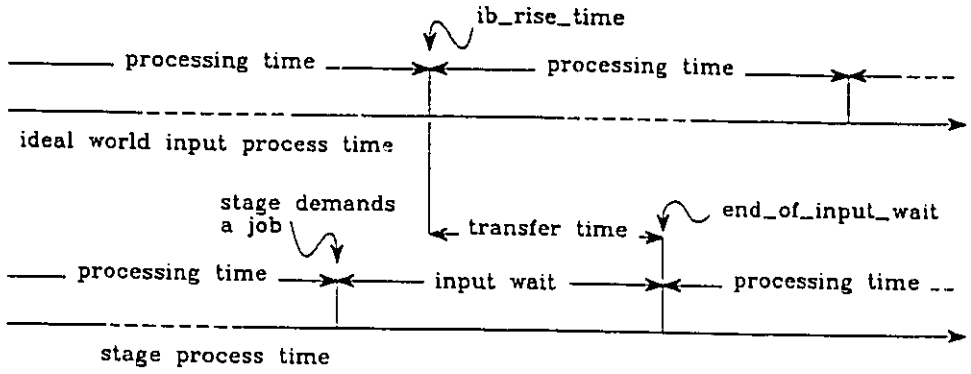


Figure 4.4 illustration of ideal world input wait

The point in time marking the transition of the ideal world input storage from empty to having one job is called (ib rise time), which refers to the start cycle time of the stage:

$$\text{ib_rise_time} = (2 - \text{`ib}) \times \text{mct} \qquad 4.15$$

proof: The definition of the input storage content of an element includes also the job currently being processed by the stage of the element. The input storage content therefore must be equal to two at the time when it has exactly one job outside of the stage ready to be transferred to the stage. Both ('ib) and (ib rise time) refer to the start cycle time of the stage and the 'ideal world input' works at a rate of 1/mct. [*EndOfProof*]

A negative (ib rise time) means the described transition will not happen during the current cycle. (End of input wait) on the bottom diagram of figure 4.4 marks the

time when a job reaches the stage in the element:

$$\text{end_of_input_wait} = \text{ib_rise_time} + \text{input_transfer_time} \qquad 4.16$$

proof: At (ib rise time) the input storage of the element has exactly one job, which is coming from the 'ideal world input', to be transferred to the stage. The 'input transfer time' is the time it takes for the job to reach the stage. [*EndOfProof*]

An input wait will occur if the stage demands a job before the time marked by (end of input wait). Therefore, an ideal world input wait will happen if (ct < end_of_input_wait). *r*input_wait (raw input wait) is defined as:

$$r\text{input_wait} = \text{end_of_input_wait} - \text{ct} \qquad 4.17$$

The ideal world input wait duration is then:

$$\text{input_wait} = \text{pos}(\, r\text{input_wait}\,) \qquad 4.18$$

The update of ib now becomes:

$$\text{ib} = \begin{cases} \text{ib_limit} & \text{if input_delay} > 0 \\ 1 + \dfrac{\text{input_transfer_time}}{\text{mct}} & \text{if input_wait} > 0 \\ \text{`ib} + \dfrac{\text{ct}}{\text{mct}} - 1 & \text{otherwise} \end{cases} \qquad 4.19$$

proof: Assuming input_wait > 0, the input storage content is two at (ib rise time) when a job is ready to be transferred to the stage. The input storage content is decreased by one when the stage finishes the current job. The stage is idle after it finishes the current job and before the time (end of input wait). The input storage content is therefore increased by (input_transfer_time / mct) during this idling period. This amount of

increase is added to (ib) so that the value of (ib) will correspond to the next start cycle time of the stage. [*EndOfProof*]

4.3 CHANGES IN IDEAL WORLD STORAGE CONTENT WHEN WAITING

For simplicity's sake, the formulations of input delay, input wait and the update of (ib) have been developed independent of output delay, output wait and the update of (ob) in the ideal world, throughout the above analysis. This section presents the inter-dependencies which exist among these variables and the expressions of adjustments caused by these inter-dependencies.

4.3.1 Ideal World Input Wait and Output Storage Content

The effect of an ideal world input wait on the output storage content of an element is discussed in this section. The 'ideal world input' and 'ideal world output' are always processing jobs unless they are delayed. Therefore, the 'ideal world output' is consuming the content of the output storage during the period of an input wait, unless it is delayed. This effect has been ignored by the ideal world input wait derivation. If an output wait does not occur at the same cycle, then the output storage content is changed by an input wait as follows:

$$ob = `ob - input_wait/mct \qquad 4.20$$

If both the output wait and input wait are non-zero in the same cycle, the output storage content is changed by:

$$ob = `ob - (input_wait - output_wait)/mct \qquad 4.21$$

because the output storage is already full during the period of output wait.

Another situation is when the ideal world input wait and output delay of an element are both non-zero in a cycle. 'Output delay' has an effect on the output storage in this case due to the non-zero output transfer time of the element. Figure 4.5 illustrates this situation. In figure 4.5, the vertical broken line on the far left marks the time the stage finishes a job, which is when an input wait can start. The vertical broken line on the far right marks the time the job reaches the 'ideal world output', which ends an ideal world output delay. During this period of time, which equals the output transfer time of the element, the time segment which is covered by both the input wait and the output delay is called 'overlap'. The end point of the input wait and the starting point of the output delay depends on the status of the input and output storage contents, respectively.

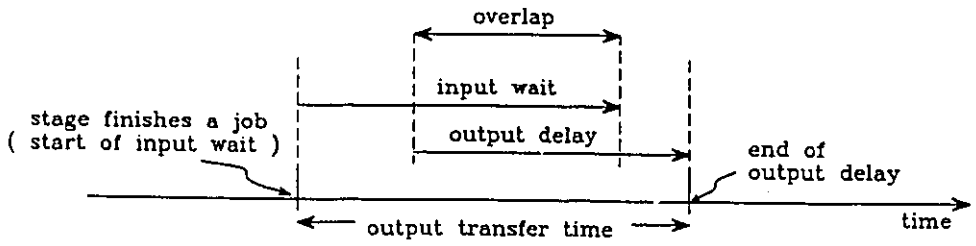


Figure 4.5 Illustration of overlap between input wait and output delay

In figure 4.5, both the input wait and the output delay are less than the output transfer time. Therefore, as is shown in the diagram, the overlap is less than the output transfer time. During the 'overlap' of the input wait and the output delay of an element, the 'ideal world output' is not consuming the output storage content because the output storage is still empty during this period of time. The occurrences of the output delay and

output wait are exclusive. Therefore, the update of the output storage content in this case is:

$$ob = `ob - (input_wait - overlap)/mct \qquad 4.22$$

Now the ideal world output storage content does not necessarily have the lower limit of one. In fact, it could become negative. The consequence of a negative (ob) is taken care of in the ideal world measurements of the next cycle.

Figure 4.6 illustrates the situation where both the input wait and output delay are longer than the ideal world output transfer time. The 'overlap' in this case takes its maximum value which is equal to the output transfer time of the element. In the same way, the value of 'overlap' can be determined for any values the 'ideal world input wait' and 'ideal world output delay' may have.

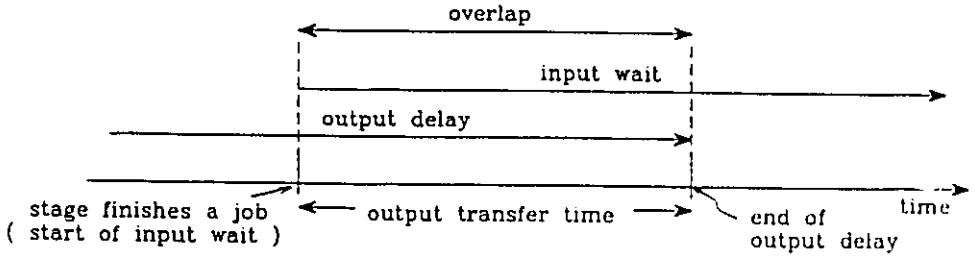


Figure 4.6 Illustration of overlap between input wait and output delay, II

4.3.2 Ideal World Output Wait and Input Storage Content

The effect of an ideal world output wait on the input storage content of an element is discussed in this section. The 'ideal world input' is filling the input storage during an

output wait unless it is delayed. This effect has been ignored by the ideal world output wait derivation. The transfer times of an element is not a factor in this case. If an input wait of the same cycle is non-zero, then the input storage content is changed by an output wait as shown below:

$$ib = \min(ib_limit, `ib + \max(\frac{output_wait - input_wait}{mct}, 0)) \qquad 4.23$$

The input storage content after an input wait has occurred, is updated to the value of the next start cycle time, as previously discussed. Therefore, (input wait) which overlaps (output_wait) is subtracted from the (output_wait) in equation 4.23. The upper bound (ib_limit) is forced in equation 4.23 to the value of (ib), because the increase of (ib) due to (output wait) affects the ideal world measurements of only the current cycle. A non-zero output wait of an element increases the amount of time the 'ideal world input' is blocked, which is (input delay). Therefore, input delay of an element can now be defined as:

$$input_delay = pos(`input_delay + output_wait) \qquad 4.24$$

4.4 IDEAL WORLD OPERATION TIME

The operation time of an element in the ideal world is discussed in this section. The ideal world operation time of an element is initially zero. It is updated after each processing cycle is finished and the ideal world measurement is calculated. The ideal world operation time is updated from the current start cycle time to the next start cycle time. The start cycle time of an element is called (Sct):

$$Sct = `Sct + ct + \max(\text{input_wait}, \text{output_wait}) \qquad 4.25$$

where the cycle time and the input and output waits are those of the current cycle.

proof: The input and output waits of an element start at the same time at the end of the current cycle, and overlap each other. Therefore, the maximum of the input and output waits plus the cycle time is the total time a stage spends from the current start cycle time to the next start cycle time. [*EndOfProof*]

4.5 CONCLUSION

In this chapter the independent measurement of an element has been defined. The calculations of independent measurements under the ideal world have been derived. The ideal world has been formulated based on the assumptions that the system is 'notionally' balanced, and the requirement that the independent measurements be consistent with time and with different elements. The 'ideal world input' and the 'ideal world output' process jobs at a constant rate, which is set to equal the mean processing rate of the stage of the element being measured. The effect of the cycle time variations of a stage in an element can then be measured by an ideal world.

The ideal world measurement is designed to keep an element in a hypothetical environment isolated from other elements of the system. Therefore, the updating of the variables such as the storage contents of an element and the ideal world operation time, are different in the ideal world than in the real world. Exact calculations of the ideal world parameters for an element, such as the storage contents and operation time, and the ideal world measurements, such as input delay and output wait, have been derived.

The ideal world measurements are the basis of the methodology. The calculations presented in this chapter for the ideal world parameters and the ideal world measurements have been tested with numerous emulations by assuming integer values for the storage sizes and initial storage contents. The case where real number values for storage sizes and initial storage contents are used cannot be tested by emulation but the correctness of the ideal world calculations can be verified when these ideal world parameters and measurements are used to construct the real world parameters.

The ideal world calculations presented in this chapter were complicated by the assumption of non-zero transfer times. There were two reasons for this assumption. Firstly, transfer times are significant when compared with the stage processing times. In a realistic observation, the transfer times are no less than 1/5 of the stage processing times. Secondly, although transfer times are only a fraction of the stage processing times, their subsequent effect are unknown in advance. Therefore, in order for the subsequent calculations to be exact, the detail results of having non-zero transfer times must not be ignored.

In summary, the ideal world calculation procedure for an element has been outlined in the following block diagram. This procedure is performed for an element at the end of every cycle using the current cycle time. The number inside a bracket in the diagram indicates the section of the chapter which describes the operation.

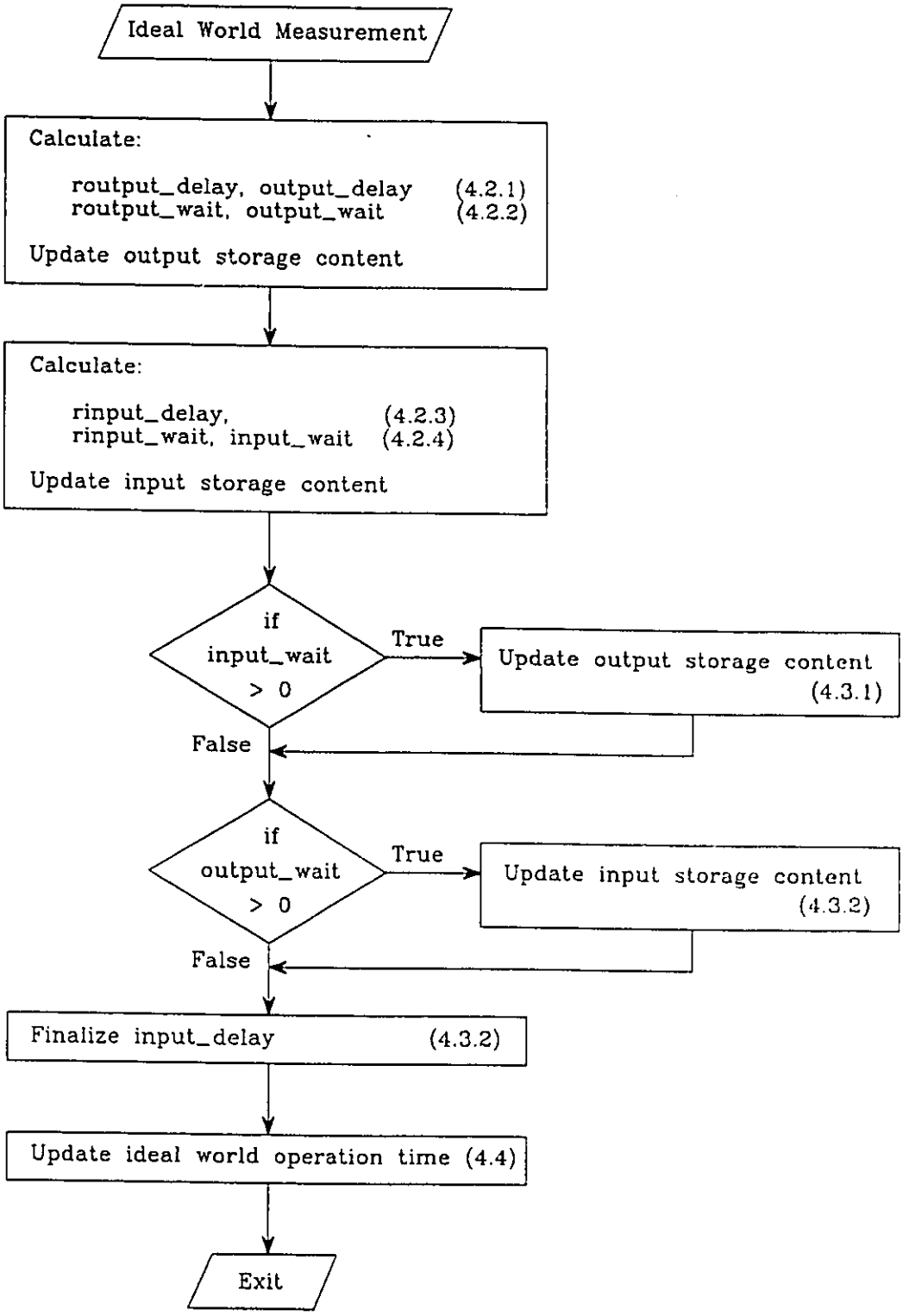


Figure 4.7 Block diagram for the ideal world calculation procedure

CHAPTER 5

CONSTRUCTING THE REAL WORLD ENVIRONMENT

5.1 INTRODUCTION

The purpose of this chapter is to derive the relationship between the ideal world measurements of each element and the real world system parameters. The ideal world measurement described in chapter four is the stand alone performance of an element where an element contains a stage and its surrounding storage. The real world parameters include among others, the duration of the blocking and starvation of any stage and the real world operating time. The system parameters are affected by the inter-dependencies of all of the elements when the elements are connected to form a system.

Using the results of this chapter, the abstract elements are then sufficient to represent a system in the real world. The abstract representation, the ideal world measurement and the relationships derived in this chapter form the 'frame work' of the current methodology. This 'frame work' is the basis for inferring knowledge, and reasoning about a system in the real world. The remaining part of this section will discuss some of the concepts involved in the calculations of the system parameters in the real world.

5.1.1 Notation Conventions

For the rest of the thesis, a consistent notation will be used to define variables. A variable name is constructed with three parts as follows:

1. a "PREFIX NOTATION", say $pE_{k_}$, where

   p = i — if the variable is considered in the 'ideal' world.

   = nothing — if the variable is considered in the 'real' world.

   E = short for 'Element'

   k = [subscripted] represents the element index.

   _ = binding symbol to either a variable name or any entity (typically a device) which belongs to the corresponding element.

2. a variable name notation as defined in the Nomenclature or in the text.

3. an optional cycle index which is always subscripted, say (j).

Example:

iE_{i+1}_total_output_delay$_{j-1}$ is the variable 'total_output_delay' associated with 'Element' 'i+1' in the 'ideal world' of cycle 'j-1'.

iE_i_ideal_world_input is the device 'ideal_world_input' associated with element 'i' considered in the 'ideal world'.

E_{k-5}_output_wait is the variable 'output_wait' belonging to the 'Element' 'k-5' in the 'real world'. Since there is no cycle index, this notation implies 'for all cycles'.

Job Flow:

The total number of elements in a serial system is n. An element in the system

is distinguished by the subscripted index, such as element$_i$ (i=1..*n*). The direction of job flow in a system is from element$_i$ to element$_{i+1}$. Figure 5.1 shows two neighbouring elements.

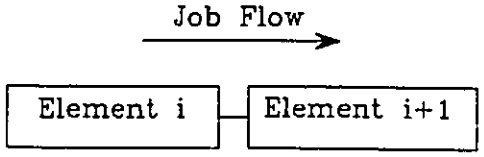


Figure 5.1 Conventions for a two element sub-system

5.1.2 The 'Shifting' of an Element

If the experience of a stage in an element in the ideal world is not exactly the same as that in the real world, then the element is 'shifted'. For an element not to be 'shifted' under the current assumptions, the only requirement is that every wait experienced by a stage in the ideal world is exactly the same as that experienced in the real world. An element being 'shifted' is illustrated below in figure 5.3.

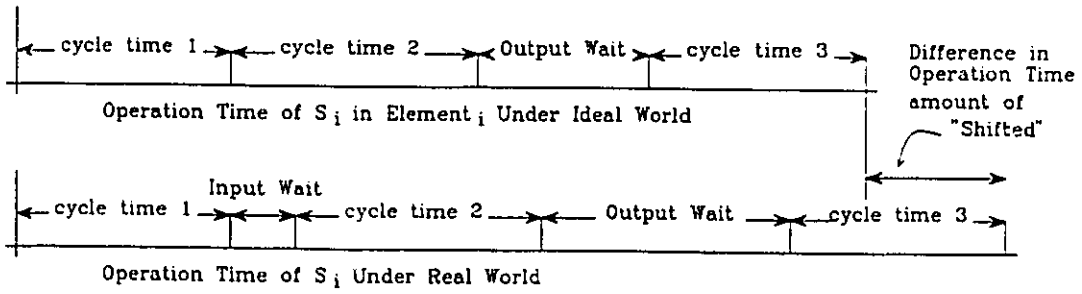


Figure 5.2 Illustration of the 'shifting' of an Element

There are two causes of the 'shifting' illustrated in figure 5.3. The first is the

input wait which occurred at the end of the first cycle in the real world, but did not occur in the ideal world. The second is the output wait which occurred at the end of the second cycle in the real world, as it also did in the ideal world but with a different duration period.

The amount an element is 'shifted' can be defined as the difference between the operation times of the ideal world and those of the real world:

$$\text{shifted} = \text{ideal world operation time} - \text{real world operation time} \qquad 5.1$$

The amount an element is 'shifted' has an initial value of zero since the operations of a stage in both worlds start simultaneously at time zero.

5.1.3 The Stand Alone Operation Time of an Ideal World

The stand alone operation time of an ideal world refers to the operation time of an 'ideal world output' or an 'ideal world input' when it is not connected to an element. Consider the 'ideal world output' of an element where the processing time is fixed at the mean cycle time (mct) of a balanced serial system. There is an infinite amount of output space inside an 'ideal world output' according to its definition. The stand alone operation timing diagram for the 'ideal world output' looks like this:

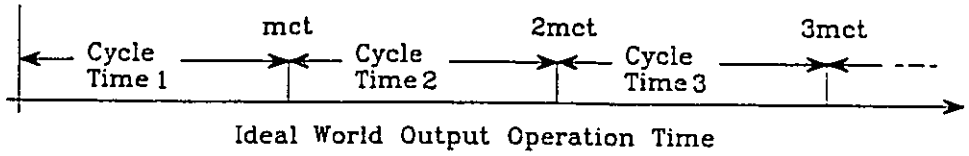


Figure 5.3 Stand alone ideal world output operation

When the 'ideal world output' is connected to an element, the operation timing diagram would be as shown in figure 5.5, where the ideal world output delay is caused by the element connected to it. The ideal world output delay has been described in chapter four.

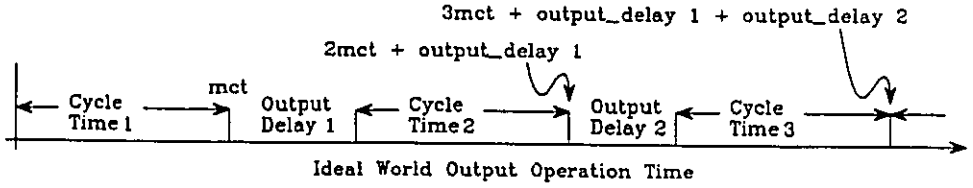


Figure 5.4 Ideal world output operation when connected to an element

The start cycle time of an 'ideal world output' when standing alone is an integer (N_{out}) multiple of the mean cycle time where (N_{out}) is the number of jobs being processed by the 'ideal world output'. However, this is not true when the 'ideal world output' is connected to an element. If the start cycle time of an 'ideal world output' is defined as (*so*Sct) when it is standing alone, and (oSct) when it is connected to an element, say for example, $element_i$, and $element_i$ is currently processing $cycle_j$, then:

$$\begin{aligned} iE_i_soSct &= iE_i_oSct - iE_i_total_output_delay_{j-1} \\ &= iE_i_N_{out} \times mct \end{aligned} \qquad 5.2$$

Likewise, if the start cycle time of an 'ideal world input' is defined as (siSct) when it is standing alone, and (iSct) when it is connected to an element, say for example, $element_{i+1}$, and $element_{i+1}$ is currently processing $cycle_k$, then:

$$\begin{aligned} iE_{i+1}_siSct &= iE_{i+1}_iSct - iE_{i+1}_total_input_delay_{k-1} \\ &= iE_{i+1}_N_{in} \times mct \end{aligned} \qquad 5.3$$

where N_{in} is the number of jobs being processed by the 'ideal world input'. Generally, the stand alone operation times of an 'ideal world output' and an 'ideal world input' can be divided by (mct) to obtain the number of jobs processed so far by the two devices.

Finally, in the ideal world measurement setting for an element, the operation time of an 'ideal world input', 'ideal world output' and the element connected to them must be the same, since they are under the same (ideal) world. This is the principle of conservation of time.

5.2 REAL WORLD OUTPUT WAIT

A real world output wait between two stages is the amount of time which S_i spends waiting for a space to output a finished job because the storage between S_i and S_{i+1} is full. This amount of time is the difference between the S_i end of cycle time and the S_{i+1} end of cycle time after the storage is full.

Consider a two stage sub-system. A real world output wait occurs when S_i is producing relatively faster than S_{i+1}. If one of the two stages is producing at a 'normal' speed, then output wait can happen if S_i is producing faster than it is supposed to or S_{i+1} is producing slower than it is supposed to. However, this statement needs to be made

with reference to 'normal' speed. In this methodology, this reference is the ideal world measurement.

Consider the case when an output wait has occurred because S_i is producing too quickly in relation to the ideal world. This is indicated by a non-zero ideal world output wait, or equivalently, a positive (iE_i_output_wait) of element$_i$, as described in chapter four. If at this moment, the storage between S_i and S_{i+1} is full and S_{i+1} has been producing at a rate exactly equal to that of the 'ideal world output' of element$_i$, then the real world output wait experienced by S_i would exactly equal iE_i_output_wait.

However, if S_{i+1} has been producing at a rate different to that of the 'ideal world output' of element$_i$, and if the difference between the times when an output storage space becomes available for the current cycle of S_i in the real world and when it becomes available in the ideal world is 'iE_{i+1}_late', then the 'real world output wait' should become (iE_i_output_wait + iE_{i+1}_late) where iE_{i+1}_late is positive if S_{i+1} has been slower than the 'ideal world output' of element$_i$, and negative otherwise. The transfer time between two stages has no effect on the real world output wait. A real world output wait has been illustrated in figure 5.6.

The current expression for real world output wait will only be true if element$_i$ and element$_{i+1}$ have not been 'shifted' as defined in section 5.1.2. However, if element$_i$ has been positively shifted before the start of the current processing cycle, cycle j, then the real world operation timing diagram for element$_i$ would be to the left of that shown at the top of figure 5.6. The real world output wait is then increased by E_i_shifted$_{j-1}$, the amount which element$_i$ has been 'shifted'. On the other hand, if element$_{i+1}$ has been positively shifted before the start of the current processing cycle, cycle k, then the real

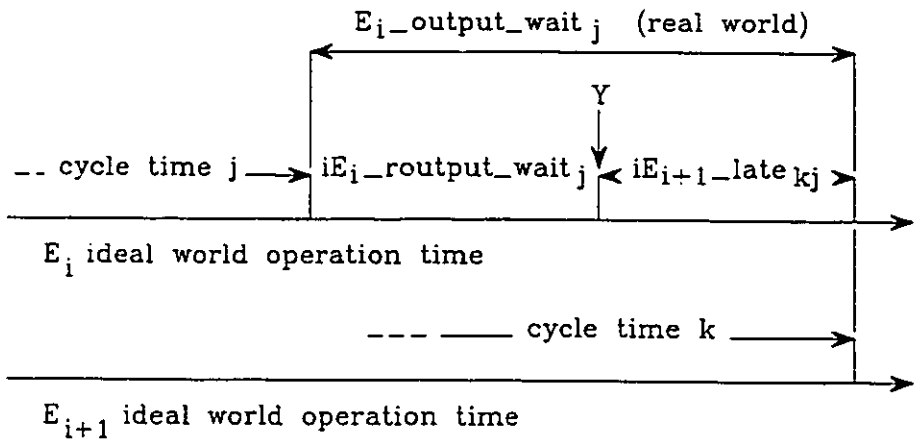


Figure 5.5 Real World Output World

world operation timing diagram of element$_{i+1}$ would also be to the left of that shown in the bottom of figure 5.6. The real world output wait is then decreased by $E_{i+1}_shifted_{k-1}$, the amount element$_{i+1}$ has been 'shifted'. Therefore, the real world output wait expression for cycle j of stage$_i$ and cycle k of stage$_{i+1}$ in general is:

$$E_i_output_wait_j = iE_i_routput_wait_j + E_i_shifted_{j-1} + iE_{i+1}_late_{kj} - E_{i+1}_shifted_{k-1} \qquad 5.4$$

If a real world output wait between stage$_i$ and stage$_{i+1}$ has occurred, its duration can be calculated using the above equation. However, the derivation of the real world output wait will not be complete if the condition describing the occurrence of an output wait is not derived. This condition requires that both stages have jobs to produce and the storage space between the two stages in the real world to be full. In this methodology, this condition should be expressed in terms of the ideal world measurements.

5.2.1 Condition for Real World Output Wait

As stated in chapter four, (iE_{i+1}_ib_full_time$_k$) is the time it takes to block the 'ideal world input' of element$_{i+1}$, with reference to the ideal world start cycle time of the current cycle, cycle k. (iE_{i+1}_sib_full_time$_k$) is defined to be the corresponding time on the stand alone time scale of the element's 'ideal world input' which has been defined in section 5.1.3. If (iE_{i+1}_sib_full_time$_k$) is an absolute value (non-relative), meaning that it refers to time zero of its time scale, then under the conservation of time:

$$iE_{i+1}_sib_full_time_k = iE_{i+1}_Sct_k + iE_{i+1}_ib_full_time_k - iE_{i+1}_total_input_delay_{k-1} \quad 5.5$$

where iE_{i+1}_Sct$_k$ is the start cycle time for cycle$_k$ of stage$_{i+1}$ in element$_{i+1}$ under the ideal world. Notice that the cycle index k for stage$_{i+1}$ is not the same as N_{in} as defined in 5.1.3. (iE_{i+1}_sib_full_time$_k$) is the projection of (iE_{i+1}_ib_full_time$_k$) on the stand alone time scale of the element's 'ideal world input'.

Also as stated in chapter four, (iE_i_ob_fall_time$_j$) is the time it takes for the output storage of element$_i$ to decrease from full to having one space, with reference to the start cycle time of the current cycle, cycle j. (iE_i_ob_fall_time$_j$) is positive if the element's output storage is indeed full at the start cycle time and negative otherwise. (iE_i_sob_fall_time$_j$) is defined to be the corresponding time on the stand alone time scale of the element's 'ideal world output', which has been defined in section 5.1.3. If (iE_i_sob_fall_time$_j$) is an absolute value, meaning that it refers to time zero of its time scale, then under the conservation of time:

$$iE_i_sob_fall_time_j = iE_i_Sct_j + iE_i_ob_fall_time_j - iE_i_total_output_delay_{j-1} \quad 5.6$$

The cycle index j for $stage_i$ is not the same as N_{out} which has been defined in 5.1.3.

($iE_i_sob_fall_time_j$) and ($iE_{i+1}_sib_full_time_k$) are illustrated in figure 5.7 using the stand alone time scales of the 'ideal world output' and the 'ideal world input' of $element_i$ and $element_{i+1}$, respectively.

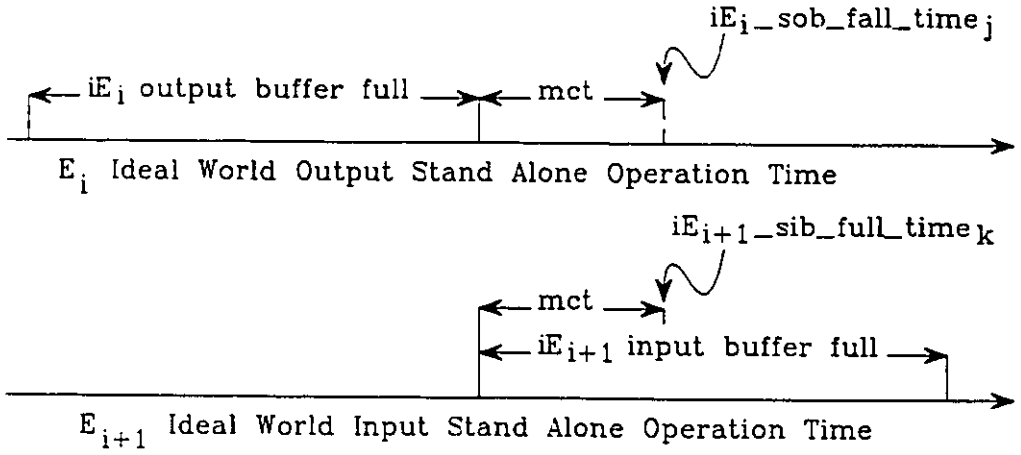


Figure 5.6 Regions of output/input buffer full for $element_i$/$element_{i+1}$, respectively

The condition under which a real world output wait occurs for cycle j of $element_i$ and cycle k of $element_{i+1}$ is if the following is true:

$$iE_i_sob_fall_time_j = iE_{i+1}_sib_full_time_k \quad 5.7$$

proof:

Define:

S The number of initial empty storage spaces in the real world between $stage_i$ and $stage_{i+1}$

N_i The number of jobs finished by $stage_i$ since the beginning

s_{out} The number of initial empty output storage spaces in an element

s_{in} The number of initial empty input storage spaces in an element

N_{in} The number of jobs finished by the 'ideal world input' of an element

N_{out} The number of jobs finished by the 'ideal world output' of an element

These parameters are illustrated in figure 5.8.

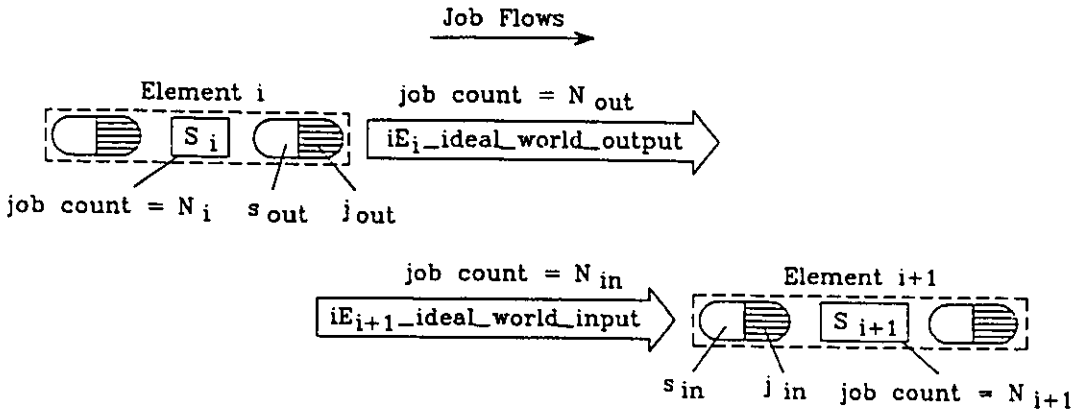


Figure 5.8 Job counts and initial conditions for $element_i$ and $element_{i+1}$

The following are true:

$$N_i \le N_{i+1} + S \qquad 5.8$$

$$iE_{i+1}_N_{in} \le N_{i+1} + iE_{i+1}_s_{in} \qquad 5.9$$

$$N_i \le iE_i_N_{out} + iE_i_s_{out} \qquad 5.10$$

$$iE_i_s_{out} + iE_{i+1}_s_{in} = S \qquad 5.11$$

Equations 5.8 to 5.10 exist due to the fact that a stage is blocked when the storage space

at its output is full in either the ideal world or the real world. Equation 5.11 is one of the rules used for setting up the elements. The following are deduced:

$$N_{i+1} + S = iE_{i_}N_{out} + iE_{i_}s_{out} \quad 5.12$$

$$N_i - S = iE_{i+1_}N_{in} - iE_{i+1_}s_{in} \quad 5.13$$

The condition under which a real world output wait can occur is:

$$N_i = N_{i+1} + S \quad 5.14$$

According to equations 5.12, 5.13 and 5.14, the following must be true:

$$iE_{i_}N_{out} = iE_{i+1_}N_{in} \quad 5.15$$

$$iE_{i+1_}N_{in} = N_{i+1} + iE_{i+1_}s_{in} \quad 5.16$$

$$N_i = iE_{i_}N_{out} + iE_{i_}s_{out} \quad 5.17$$

It can be observed that satisfying equations 5.15, 5.16 and 5.17 is equivalent to satisfying equation 5.14, which is the condition under which a real world output wait can occur. Equation 5.15 is satisfied by equation 5.7 because after both sides of equation 5.7 are divided by (mct), the results are $iE_{i_}N_{out}$ and $iE_{i+1_}N_{in}$ respectively on each side, according to the definition of stand alone operation times of the 'ideal world output' and 'ideal world input'. Equations 5.16 and 5.17 indicate that the input storage of $element_{i+1}$ and output storage of $element_i$ are full, respectively under the ideal world. In order to investigate the time regions where $iE_{i+1_}ib$ and $iE_{i_}ob$ are full, $iE_{i+1_}N_{in}$ is renamed $iE_{i+1_}N_{in_}full$ when it satisfies equation 5.16, $iE_{i_}N_{out}$ is renamed $iE_{i_}N_{out_}full$ when it satisfies equation 5.17. According to the definition contained in section 4.2.2,

(iE_i_ob_fall_time$_j$) is the instant marking the transition of the output storage of an element when it changes from full to having just enough space for one job and according to the definition that the stand alone operation time of 'ideal world output' is equal to (N_{out} x mct):

$$iE_i_N_{out}_full \times mct \leq iE_i_ob_fall_time_j - mct \qquad 5.18$$

According to the definition contained in section 4.2.3, (iE_{i+1}_ib_full_time$_k$) is the instant when the input storage of an element is full and the 'input ideal world' becomes ready to transfer a job and according to the definition that the stand alone operation time of 'ideal world input' is equal to (N_{in} x mct):

$$iE_{i+1}_ib_full_time_k - mct \leq iE_{i+1}_N_{in}_full \times mct \qquad 5.19$$

The situations described by equations 5.18 and 5.19 are shown in figure 5.7. Following equations 5.8, 5.16 and 5.17 the following must be true:

$$iE_i_N_{out}_full \leq iE_{i+1}_N_{in}_full \qquad 5.20$$

In order to satisfy equations 5.15, 5.16 and 5.17, when constrained by equations 5.18 5.19 and 5.20, equation 5.7 must be true in order for a real world output wait to happen. [*EndOfProof*]

5.2.2 Expressing iE_{i+1}_late$_{kj}$ In Terms of Ideal World Measurements

The definition of iE_{i+1}_late$_{kj}$ is obtained from the derivation of E_i_output_wait in the situation where both element$_i$ and element$_{i+1}$ have not been 'shifted'. Therefore, the derivation of iE_{i+1}_late$_{kj}$ will proceed in the same situation where both element$_i$ and

$element_{i+1}$ have not been 'shifted'. According to the definition of the 'shifting' of an element, this situation occurs where both the ideal world operation times of $element_i$ and $element_{i+1}$ are equal to the real world operation time.

Point Y in figure 5.6 is equal to ($iE_i_Sct_j$ + $iE_i_ob_fall_time_j$), which is the end of ($iE_i_output_wait_j$). Defining (Ect) as the end of the processing cycle time of an element:

$$iE_{i+1}_late_{kj} = iE_{i+1}_Ect_k - (iE_i_ob_fall_time_j + iE_i_Sct_j) \qquad 5.21$$

However, it is desirable to express $iE_{i+1}_late_{kj}$ in terms of the ideal world input delay of $element_{i+1}$. $iE_i_ob_fall_time_j$ is expanded upon using equation 5.6:

$$iE_{i+1}_late_{kj} = iE_{i+1}_Ect_k - iE_i_sob_fall_time_j - iE_i_total_output_delay_{j-1} \qquad 5.22$$

Using the equality of equation 5.7 for processing cycles j and k of element i and i+1 respectively, and expanding $iE_{i+1}_sob_fall_time_k$ using equation 5.5, then:

$$\begin{aligned} iE_{i+1}_late_{kj} = {} & iE_{i+1}_Ect_k - iE_i_total_output_delay_{j-1} - iE_{i+1}_Sct_k \\ & - iE_{i+1}_ib_full_time_k + iE_{i+1}_total_input_delay_{k-1} \end{aligned} \qquad 5.23$$

recognizing that for cycle k of $element_{i+1}$:

$$iE_{i+1}_rinput_delay_k = iE_{i+1}_Ect_k - (iE_{i+1}_Sct_k + iE_{i+1}_ib_full_time_k) \qquad 5.24$$

then:

$$\begin{aligned} iE_{i+1}_late_{kj} = {} & iE_{i+1}_rinput_delay_k - iE_i_total_output_delay_{j-1} \\ & + iE_{i+1}_total_input_delay_{k-1} \end{aligned} \qquad 5.25$$

As long as equation 5.7 is true for cycles j and k of elements i and i+1 respectively, the expression for real world output wait can be verified. Combining

equation 5.4, which is the definition of the real world output wait, equation 5.25 which is the definition of $iE_{i+1}_late_{kj}$, equations 5.5 and 5.6 and expanding upon $iE_i_routput_wait_j$ and $iE_{i+1}_rinput_delay_k$ using their definitions which are contained in chapter four, after simplification:

$$E_i_output_wait_j = (- iE_i_Ect_j + E_i_shifted_{j-1}) + (iE_{i+1}_Ect_k - E_{i+1}_shifted_{k-1}) \qquad 5.26$$

According to equation 5.1, the definition of 'shifted', equation 5.26 is equivalent to:

$$E_i_output_wait_j = E_{i+1}_Ect_k - E_i_Ect_j \qquad 5.27$$

which is the definition of the real world output wait between $stage_i$ and $stage_{i+1}$.

5.3 REAL WORLD INPUT WAIT

A real world input wait between two stages is the amount of time S_i spends waiting for a job to start a new processing cycle after finishing the current job because the storage between S_{i-1} and S_i is empty. This amount of time is the difference between the S_i end of current cycle time and the S_{i-1} end of current cycle time plus the transfer time between the two stages, after the storage is empty.

Consider a two stage sub-system. A real world input wait can happen if S_i is producing relatively faster than S_{i-1}. As with the case of a real world output wait, the ideal world measurement is used as a reference for the relative speeds of the two stages. The calculation of the real world input wait from the ideal world measurement is very similar to that of the real world output wait. When $element_{i-1}$ and $element_i$ of $stage_{i-1}$ and $stage_i$ are not 'shifted', the length of a real world input wait is ($iE_i_rinput_wait$ +

iE_{i-1}_late) where (iE_{i-1}_late) is the difference between the times when a job becomes available to $stage_i$ in the real world and when it becomes available in the ideal world. A real world input wait is illustrated in figure 5.9:

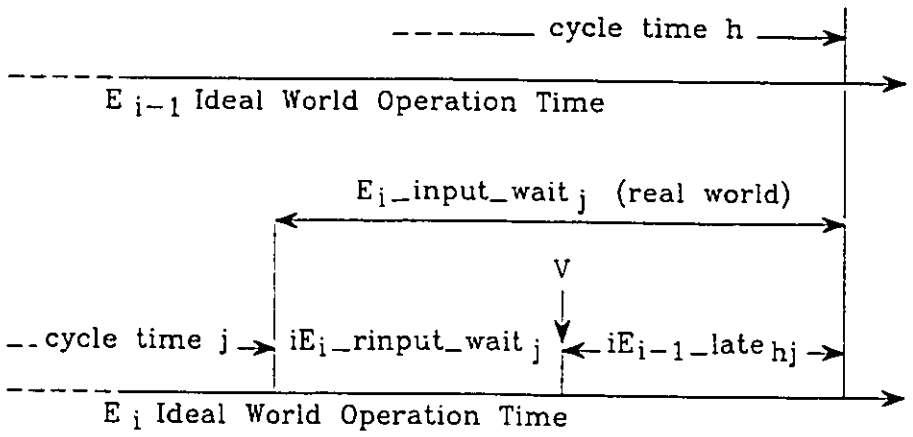


Figure 5.8 Real World Input Wait and Ideal World Measurement

When the elements are 'shifted', the general real world input wait expression for cycle h of $stage_{i-1}$ and cycle j of $stage_i$ is:

$$E_i_input_wait_j = \quad 5.28$$
$$(iE_{i-1}_late_{hj} - E_{i-1}_shifted_{h-1}) + (iE_i_input_wait_j + E_i_shifted_{j-1})$$

For example, if $element_{i-1}$ has shifted positively previous to processing cycle h, then the real world operation timing diagram of E_{i-1} would appear to the left of that shown at the top of figure 5.8. The real world input wait is then decreased by iE_{i-1}_$shifted_{h-1}$ which is the amount E_{i-1} has 'shifted'. On the other hand, if $element_i$ has positively shifted previous to processing cycle j, then the real world operation timing diagram of E_i would also be shifted to the left of that shown at the bottom of figure 5.8. The real world input

wait is then increased by $iE_i_shifted_{j-1}$ which is the amount E_i has 'shifted'.

The condition for having a real world input wait between $stage_{i-1}$ and $stage_i$ is that the storage space between them is empty when both stages have jobs to produce. This condition will be expressed in terms of the ideal world measurements.

5.3.1 Condition for Real World Input Wait

As stated in chapter four, ($iE_{i-1}_ob_empty_time_h$) is the time it takes to starve the 'ideal world output' of $element_{i-1}$, with reference to the start cycle time of the current cycle, $cycle_h$. ($iE_{i-1}_sob_empty_time_h$) is defined to be the corresponding time on the stand alone time scale of the 'ideal world output' of the element, which has been defined in section 5.1.3. If ($iE_{i-1}_sob_empty_time_h$) is an absolute value, meaning that it refers to time zero of its time scale, then under the conservation of time:

$$iE_{i-1}_sob_empty_time_h = iE_{i-1}_Sct_h + iE_{i-1}_ob_empty_time_h - iE_{i-1}_total_output_delay_{h-1} \qquad 5.29$$

Notice that the cycle index h for $stage_{i-1}$ is not the same as N_{out} as defined in 5.1.3. ($iE_{i-1}_sob_empty_time_h$) is the projection of ($iE_{i-1}_ob_empty_time_h$) on the stand alone time scale of the 'ideal world output' of the element.

Also stated in chapter four, ($iE_i_ib_rise_time_j$) of $element_i$ is the time it would take for the input storage of the element to rise from empty to having one job, with reference to the start cycle time of the current cycle, cycle j. ($iE_i_ib_rise_time_j$) is positive if the element's input storage is indeed empty at the start cycle time and negative otherwise. ($iE_i_sib_rise_time_j$) is defined to be the corresponding time on the stand alone time scale of the 'ideal world input' of the element, which has been defined in section

5.1.3. If (iE_i_sib_rise_time$_j$) is an absolute value, meaning that it refers to time zero of its time scale, then under the conservation of time:

$$iE_i_sib_rise_time_j = iE_i_Sct_j + iE_i_ib_rise_time_j - iE_i_total_input_delay_{j-1} \quad 5.30$$

The cycle index j for $stage_i$ is not the same as N_m as defined in 5.1.3.

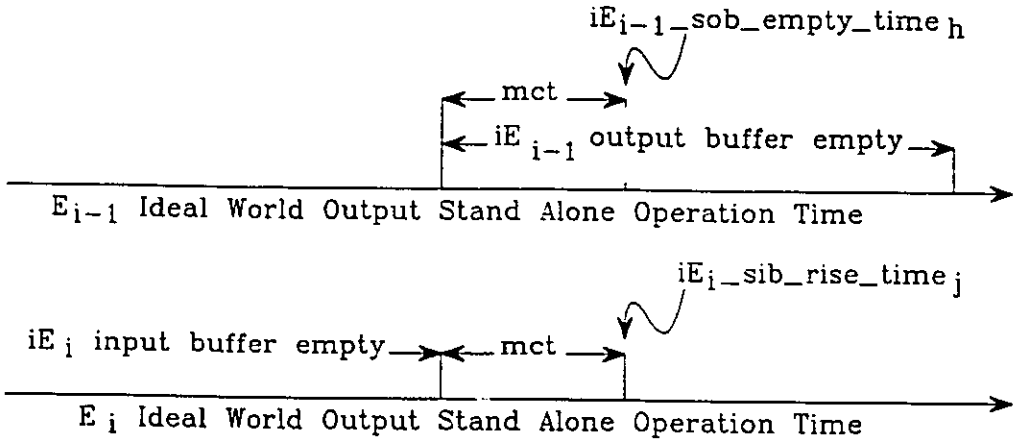


Figure 5.9 Regions of output/input buffer full for $element_i$/$element_{i+1}$, respectively

(iE_{i-1}_sob_empty_time$_h$) and (iE_i_sib_rise_time$_j$) are illustrated in figure 5.10 using the stand alone time scales of the 'ideal world output' and the 'ideal world input' of $element_{i-1}$ and $element_i$, respectively. The condition under which a real world input wait happens is if the following is true:

$$iE_{i-1}_sob_empty_time_h = iE_i_sib_rise_time_j \quad 5.31$$

proof:

Define:

J The number of initial jobs in the real world between $stage_{i-1}$ and $stage_i$

N_i The number of jobs finished by $stage_i$ since the beginning

j_{out} The number of initial jobs in the output storage in an element

j_{in} The number of initial jobs in the input storage in an element

N_{in} The number of jobs finished by the 'ideal world input' of an element

N_{out} The number of jobs finished by the 'ideal world output' of an element

These parameters are illustrated in figure 5.8 using $element_i$ and $element_{i+1}$. The following are true:

$$N_i \le N_{i-1} + J \qquad 5.32$$

$$N_i \le iE_{i-}N_{in} + iE_{i-}j_{in} \qquad 5.33$$

$$iE_{i-1-}N_{out} \le N_{i-1} + iE_{i-1-}j_{out} \qquad 5.34$$

$$iE_{i-}\, j_{in} + iE_{i-1-}\, j_{out} = J \qquad 5.35$$

Equations 5.32 to 5.34 are due to the fact that a stage is starved when the input storage space is empty in either the ideal world or the real world. Equation 5.35 is one of the rules used to set up the elements. The following is deduced:

$$N_{i-1} + J = iE_{i-}N_{in} + iE_{i-}\, j_{in} \qquad 5.36$$

$$N_i - J = iE_{i-1-}N_{out} - iE_{i-1-}\, j_{out} \qquad 5.37$$

The condition under which a real input wait occurs is:

$$N_i = N_{i-1} + J \qquad 5.38$$

The following must be true according to equations 5.36, 5.37 and 5.38:

$$iE_{i-1}_N_{out} = iE_i_N_{in} \quad 5.39$$

$$iE_{i-1}_N_{out} = N_{i-1} + iE_{i-1}_j_{out} \quad 5.40$$

$$N_i = iE_i_N_{in} + iE_i_j_{in} \quad 5.41$$

It can be observed that satisfying equations 5.39, 5.40 and 5.41 is equivalent to satisfying equation 5.31, which is the condition under which a real world input wait occurs. Equation 5.39 is satisfied by equation 5.31 because after both sides of equation 5.31 are divided by (mct), the results are $iE_{i-1}_N_{out}$ and $iE_i_N_{in}$ on each side respectively, according to the definition of the stand alone operation time of the 'ideal world output' and the 'ideal world input'. Equations 5.40 and 5.41 indicate that output storage of $element_{i-1}$ and input storage of $element_i$ are empty, respectively in the ideal world. In order to investigate the time regions where iE_{i-1}_ob and iE_i_ib are empty, $iE_{i-1}_N_{out}$ is renamed $iE_{i-1}_N_{out}_empty$ when it satisfies equation 5.40, $iE_i_N_{in}$ is renamed $iE_i_N_{in}_empty$ when it satisfies equation 5.41. According to the definition contained in section 4.2.1, ($iE_{i-1}_ob_empty_time_h$) is the time the ideal world output storage is empty and the 'ideal world output' demands a job and according to the definition that the stand alone operation time of 'ideal world output' is equal to (N_{out} x mct):

$$iE_{i-1}_sob_empty_time_h - mct \leq iE_{i-1}_N_{out}_empty \times mct \quad 5.42$$

According to the definition contained in section 4.2.4, ($iE_i_ib_rise_time_j$) is the instant marking the transition of the ideal world input storage from empty to having one job and according to the definition that the stand alone operation time of 'ideal world input' is equal to (N_{in} x mct):

$$iE_i_N_{in}_empty \times mct \leq iE_i_sib_rise_time_j - mct \qquad 5.43$$

The situations described by equations 5.42 and 5.43 are shown in figure 5.10. Following equations 5.32, 5.40 and 5.41 the following must be true:

$$iE_i_N_{in}_empty \leq iE_{i-1}_N_{out}_empty \qquad 5.44$$

In order to satisfy equations 5.39, 5.40 and 5.41, when constrained by equations 5.42 5.43 and 5.44, equation 5.31 must be true in order for an real world input wait to happen. [*EndOfProof*]

5.3.2 Expressing $iE_{i-1}_late_{hj}$ in Terms of Ideal World Measurements

The definition of $iE_{i-1}_late_{hj}$ is obtained from the derivation of $E_i_input_wait$ in the situation that both $element_{i-1}$ and $element_i$ have not been 'shifted'. Therefore, the derivation of $iE_{i-1}_late_{hj}$ will be proceeded with in the same situation where both $element_{i-1}$ and $element_i$ have not been 'shifted'. According to the definition of the 'shifting' of an element, this situation occurs when both the ideal world operation times of $element_{i-1}$ and $element_i$ are equal to the real world operation time.

Point V in figure 5.9 is equal to $(iE_i_Sct_j + iE_i_ib_rise_time_j)$ which is the end of $(iE_i_input_wait_j)$. (Ect) is defined as the end of cycle time of an element:

$$iE_{i-1}_late_{hj} = iE_{i-1}_Ect_h - (iE_i_ib_rise_time_j + iE_i_Sct_j) \qquad 5.45$$

However, it is desirable to express $iE_{i-1}_late_{hj}$ in terms of the ideal world output delay of $element_{i-1}$. Expanding on $iE_i_ib_rise_time_j$ using equation 5.30:

$$iE_{i-1}_late_{hj} = iE_{i-1}_Ect_h - iE_i_sib_rise_time_j - iE_i_total_input_delay_{j-1} \qquad 5.46$$

Using the equality of equation 5.31 for processing cycles h and j of element i-1 and i respectively, and expanding $iE_{i-1}_sob_empty_time_h$ using equation 5.29, then:

$$iE_{i-1}_late_{hj} = iE_{i-1}_Ect_h - iE_i_total_input_delay_{j-1} - iE_{i-1}_Sct_h - iE_{i-1}_ob_empty_time_h + iE_{i-1}_total_output_delay_{h-1} \quad 5.47$$

recognizing that for cycle h of $element_{i-1}$:

$$iE_{i-1}_routput_delay_h = iE_{i-1}_Ect_h - (iE_{i-1}_Sct_h + iE_{i-1}_ob_empty_time_h) \quad 5.48$$

then:

$$iE_{i-1}_late_{hj} = iE_{i-1}_routput_delay_h - iE_i_total_input_delay_{j-1} + iE_{i-1}_total_output_delay_{h-1} \quad 5.49$$

As long as equation 5.31 is true for cycles h and j of $element_{i-1}$ and $element_i$, respectively, the expression for the real world input wait can be verified. Combining equation 5.28, which is the definition of the real world input wait, equation 5.49 which is the definition of $iE_{i-1}_late_h$, equations 5.29 and 5.30 and expanding upon $iE_i_rinput_wait_j$ and $iE_{i-1}_routput_delay_h$ using their definitions which are contained in chapter four, after simplification:

$$E_i_input_wait_j = (- iE_i_Ect_j + iE_i_output_transfer_time + E_i_shifted_{j-1}) + (iE_{i-1}_Ect_h + iE_{i-1}_input_transfer_time - E_{i-1}_shifted_{h-1}) \quad 5.50$$

According to equation 5.1, which is the definition of 'shifted', equation 5.50 is equivalent to:

$$E_{i-1}_input_wait_j = E_{i-1}_Ect_h - E_i_Ect_j + Transfer_Time_{i-1,\,i} \quad 5.51$$

where $(Transfer_Time_{i-1,\,i})$ is the real world transfer time between $stage_{i-1}$ and $stage_i$,

which equals (iE_{i-1}_output_transfer_time + iE_i_input_transfer_time) according to the element setup rules. Equation 5.51 is the definition of the real world input wait for $stage_{i-1}$.

5.4 REAL WORLD PROPAGATED OUTPUT WAIT

The real world propagated output wait is the third type of wait a stage can experience in the real world. The real world output wait described in section 5.2 is between two neighbouring stages. The real world propagated output wait to be discussed in this section involves a series of consecutive stages. A propagated output wait occurs when the inter-stage storages of a series of consecutive stages are full, say from $stage_M$ to $stage_N$, and if the real world output wait happens first to one of these stages, say $stage_L$, where $M \leq i < L \leq N$ and all of the stages from $stage_i$ to $stage_{L-1}$ finish their current jobs before the output wait currently being experienced by $stage_L$ has ended, then all of the stages from $stage_i$ to $stage_{L-1}$ are blocked until the end of the output wait of $stage_L$. $Stage_i$ to $stage_{L-1}$ are all blocked by $stage_L$ in this case. In other words, the output wait has been propagated from $stage_L$ to $stage_i$.

If the length of the real world output wait between any two neighbouring stages takes positive and negative values, then the length of the propagated output wait of $stage_i$ is:

$$E_i_propagated_output_wait_j = \sum_{c=i}^{L} E_c_output_wait_{current_cycle} \qquad 5.52$$

which is simply the definition of the real world propagated output wait.

The condition under which the real world propagated output wait happens to $stage_i$ is when the conditions for the real world output wait, as described by equation 5.7, are true for all of the current cycles of $stage_c$ (c=i..L).

5.5 PERFORMANCES

This section discusses performance measurements for a stage and the system as a whole. The measurement of performances requires a reference through which production loss can be defined. The hypothetical situation of having infinite inter-stage storages is defined as the reference for the performance measurements. When inter-stage storages are infinite, the stages are said to be completely 'de-coupled', allowing each stage to operate as an independent unit (Masso, et al. 74).

When a long enough production time is given to a system with infinite inter-stage storages, the number of jobs produced by any stage is N, where (N x mct) should approximately equal the production time, since the mean processing times of all of the stages are equal to (mct). N is also the number of jobs produced by the system since the output stage of the system is an independent unit in the infinite inter-stage storage situation.

5.5.1 Ideal World Performance

Referring to section 5.1.3, equation 5.2 has been rewritten as follows for the operation time of the 'ideal world output' of an element:

$$\text{ideal_world_output_operation_time} = iE_i_N_{out} \times \text{mct} + iE_i_\text{total_output_delay}_{j-1} \qquad 5.53$$

Equation 5.3 has been rewritten as follows for the operation time of an 'ideal world input' of an element:

$$\text{ideal_world_input_operation_time} = iE_i_N_{in} \times mct + iE_i_\text{total_input_delay}_{j-1} \quad 5.54$$

Section 5.1.3 assumes that N_{out} and N_{in} are integers. Therefore, operation times are only measurable at the start cycle times. Equations 5.53 and 5.54 are extensions of equations 5.2 and 5.3 after it is assumed that N_{out} and N_{in} are real numbers so that the operation times of the 'ideal world output' and 'ideal world input' can be measured at any instant. The 'ideal world output' operation time and the 'ideal world input' operation time given by equations 5.53 and 5.54 must be equal to each other and also equal to the operation time of the stage inside the element, under the conservation of time.

Observing that the relationship between the number of jobs produced by the stage and those by the 'ideal world output' of an element:

$$iE_i_j_{out} - iE_i_ob_j + N_i = iE_i_N_{out} \quad 5.55$$

Using equations 5.53 and 5.55, the ideal world operation time of a stage is:

$$iE_i_Sct_j = (iE_i_j_{out} - iE_i_ob_j + N_i) \times mct + iE_i_\text{total_output_delay}_{j-1} \quad 5.56$$

where N_i is an integer as defined. Therefore, the operation times of a stage in an element in the ideal world are measurable only at the start cycle times. Equation 5.56 is the time stage_i requires to produce N_i jobs in the ideal world. In defining (eff_i) as the isolated efficiency of stage_i in element_i in the ideal world, then according to the reference of performance measurement:

$$eff_i = \frac{N_i \times mct}{iE_i_Sct} \qquad 5.57$$

In order for $stage_i$ to have a high (eff_i), the ideal world total output delay must be kept very low. However, this does not require the stage to work at a fixed rate equal to (mct) in order to match that of the 'ideal world output' because the storages allocated to an element allow variations in processing times. Therefore, (eff_i) would deteriorate only when the processing time variations exceed the limits provided by the given storages.

5.5.2 Real World Performance

E_i_Sct is defined as the real world operation time of $element_i$ measured at the start cycle time. According to the definition of 'shifted' of an element:

$$E_i_Sct_j = iE_i_Sct_j - E_i_shifted_{j-1} \qquad 5.58$$

Using equations 5.56 and 5.58, the real world operation time for $element_i$ is then:

$$\begin{aligned} E_i_Sct_j = {} & (iE_i_j_{out} - iE_i_ob_j + N_i) \times mct \\ & + iE_i_total_output_delay_{j-1} - E_i_shifted_{j-i} \end{aligned} \qquad 5.59$$

which is the time required for $stage_i$ of $element_i$ in the real world to produce N_i jobs. (Eff_i) is defined as the efficiency of $stage_i$ in the real world. According to the reference of performance measurement:

$$Eff_i = \frac{N_i \times mct}{E_i_Sct} \qquad 5.60$$

The efficiency of a stage in the real world is different than that in the ideal world.

The former is affected by all of the stages and storages in a system, while the latter is determined by a stage and its storages which are isolated from the rest of the system, inside an element in the ideal world. From the efficiency expressions of the two cases, it can be observed that this difference is represented by the term (E_i_shifted$_{j-1}$), which can be interpreted as the total amount of inefficiency introduced to element$_i$ by all of the other elements in the system. In a n-element system, equation 5.60 is the system efficiency expression if the subscript i is equal to n, since element$_n$ is the output element of the system and the number of jobs produced by element$_n$ is the number of jobs produced by the system.

5.6 CONCLUSION

In this chapter, the calculation of the real world parameters from the ideal world measurements has been derived. In the derivation process, the concepts of 'shifting' and the stand alone operation times for the 'ideal world output' and 'ideal world input' of an element have been defined. The parameters discussed in this chapter include waits and performances in the real world. There are three types of real world waits: the output wait, the input wait between any two neighbouring stages and the propagated output wait which involves a series of consecutive stages. These waits define the concepts of 'blocking and starvation' in a system. The real world performance of a stage or the system has been derived from the ideal world performance which has also been defined in this chapter. A hypothetical system with infinite inter-stage storages has been used as the reference for the performance measurements.

The calculation of each real world wait involves determining the duration of the

wait and the condition under which it occurs. The duration calculations of the real world waits are derived to be exact, which has been shown in the cases with real world output and input waits. The propagated output wait is a cascade of real world output waits. In calculating a real world wait, the condition for that real world wait is first evaluated. If the condition for a real world wait is not true, then a real world wait of that type is impossible under the circumstances. Otherwise, the duration of the real world wait is calculated. If the calculated real world wait is positive, then this real world wait must be equal to that of the real system.

The abstract representation of the system using elements, the ideal world measurements and the relationships between the ideal world measurements and the real world parameters derived in this chapter provide the basic 'frame work' for analysing the system. The 'frame work' follows the real system and gives an exact description of the system in abstract terms. The following block diagram describes how the 'frame work' is updated as production proceeds in the real system and summarizes the relationships which have been derived in this chapter. Assuming that the abstract representation using elements has been setup for the system and all of the variables have been properly initialized, the procedure illustrated by the block diagram is then executed for the stage, $stage_i$ (i=1..n), which has just finished its current cycle.

The two parameters inside a 'condition()' block are element indexes. There are three notes in the block diagram. The following is an expansion on the notes.

Note 1: The 'check time' block compares the calculated end of cycle time with the current real time of the system, since the procedure described by the block diagram is performed at the end of cycle time of a stage. The end of cycle time of $stage_i$ is denoted

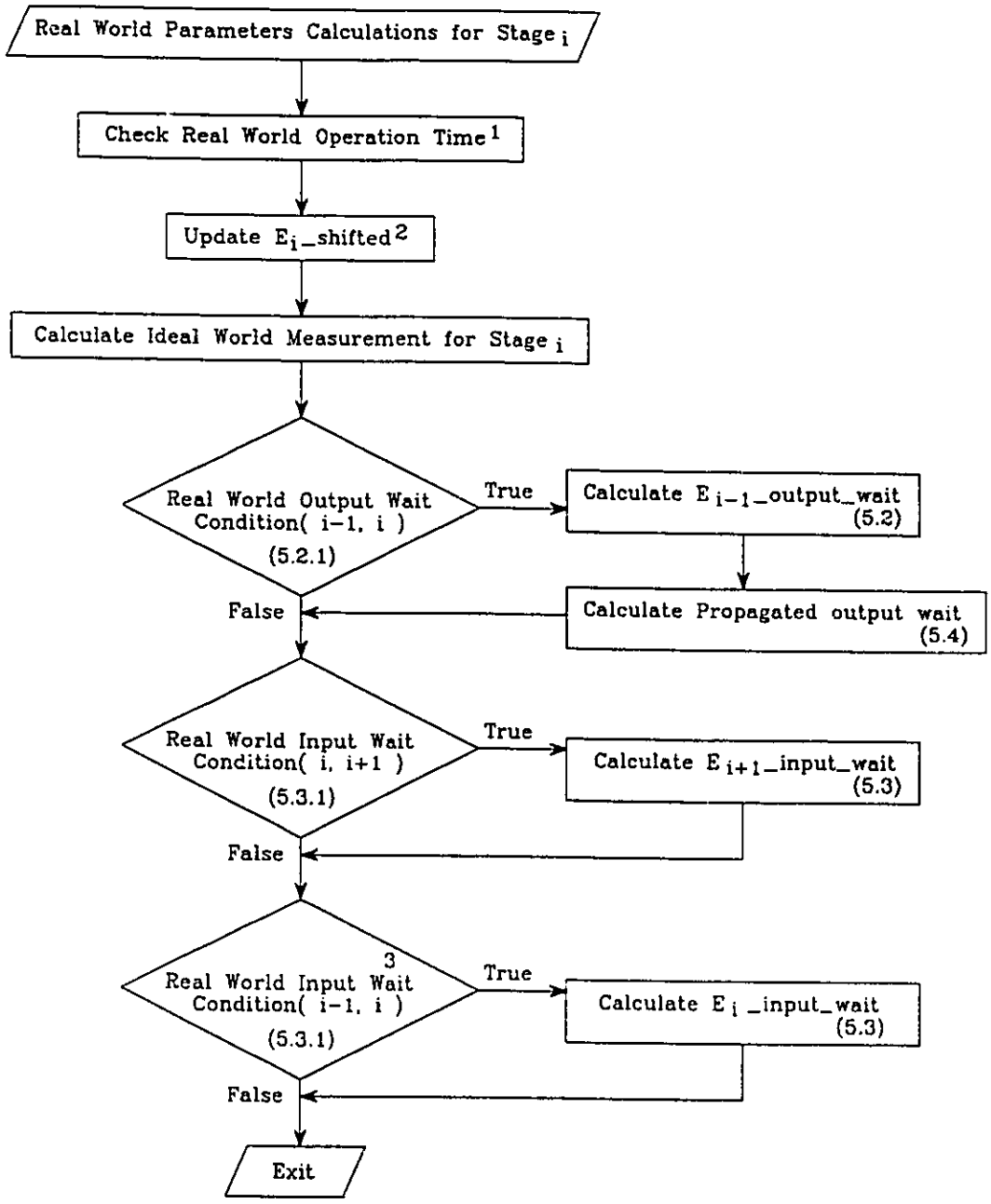


Figure 5.10 Block diagram for the real world parameter calculation procedure

by E_i_Ect and is calculated by an incremental update at the end of every cycle:

$$E_i_Ect_j = E_i_Ect_{j-1} + pos(max(E_i_output_wait_{j-1}, E_i_input_wait_{j-1})) + E_i_ct_j \quad \textbf{5.61}$$

where $E_i_Ect_0 = E_i_output_wait_0 = E_i_input_wait_0 = 0$. The checking of the calculated end of cycle time with the current real operation time is a minimum error checking procedure, which ensures that the system described by the 'frame work' follows the real system. In fact, every calculated real world parameter including among others, the end of cycle time and the real world waits are checked with that of the emulator output in all of the test cases.

Note 2: In addition to the calculation of the amount an element has been 'shifted' as defined by equation 5.1, the calculation can be alternatively done by an incremental update which is performed at the end of each cycle:

$$E_i_shifted_j = E_i_shifted_{j-1} + max(iE_i_output_wait_{j-1}, iE_i_input_wait_{j-1}) - max(E_i_output_wait_{j-1}, E_i_input_wait_{j-1}, 0) \quad \textbf{5.62}$$

where $E_i_shifted_0 = 0$. Under the assumptions of the study, the only factor which contributes to the difference between the ideal world operation time and the real world operation time is the difference between the ideal world wait and the real world wait.

Note 3: The checking of $E_i_input_wait$ at the end of cycle time of $stage_i$ is only necessary if the transfer time between $stage_{i-1}$ and $stage_i$ is non-zero. Otherwise, it is sufficient to check for $E_i_input_wait$ at the $stage_{i-1}$ end of cycle time, which ended the 'starvation' of $stage_i$. When the transfer time between $stage_{i-1}$ and $stage_i$ is non-zero, $E_i_input_wait$ can happen even though the $stage_{i-1}$ end of cycle occurred earlier than the $stage_i$ end of cycle, which started $E_i_input_wait$. $E_i_input_wait$ of this type cannot be

detected at the $stage_{i-1}$ end of cycle.

CHAPTER 6

CAUSES AND EFFECTS OF PRODUCTION LOSS

6.1 INTRODUCTION

In this chapter the results from the previous chapters are utilized for the purpose of analysing the system. The 'frame work' derived in chapter five gives an exact description of the system under the abstract representation using the 'ideal world measurement' as a reference. Based on this framework, this chapter presents a scheme which enhances the detailed information given by the description. The purpose for doing this is to provide the cause and effect information contributed by each element towards the production loss of another element, which is the basic requirement of system monitoring.

6.2 The Causes of Real World Waits Between Two Elements

In this section, an assumption will be established to quantify the causes of a real world wait by any two neighbouring elements. This assumption is based on information given by the 'frame work' which has been defined in chapter five.

The production loss of systems with finite inter-stage storages is due to the phenomena of 'blocking' and 'starvation'. The definition of production loss contained in chapter five is based on a hypothetical 'infinite inter-stage storage' reference. The

infinite inter-stage storage is an idealized situation in which the two phenomena causing production loss will not occur.

According to the terminology set out in chapter five, 'blocking' in the real world between any two stages is the 'real world output wait', and 'starvation' is the 'real world input wait'. The calculation of the occurrence and duration of these waits is based on the ideal world measurement. The ideal world measurement is the reference used to distinguish the causes of a real world wait between the two stages and also between the past and the present.

6.2.1 Real World Output Wait Causes

If an output wait has occurred, the duration is calculated using equation 5.4, which has been reproduced here:

$$E_i_output_wait_j = iE_i_routput_wait_j + E_i_shifted_{j-1} + iE_{i+1}_late_{kj} - E_{i+1}_shifted_{k-1} \qquad 6.1$$

where

$$iE_{i+1}_late_{kj} = iE_{i+1}_rinput_delay_k - iE_i_total_output_delay_{j-1} + iE_{i+1}_total_input_delay_{k-1} \qquad 6.2$$

Initially, $E_i_shifted_0$, $E_{i+1}_shifted_0$, $iE_i_total_output_delay_0$ and $iE_{i+1}_total_input_delay_0$ are zero. Then $E_i_output_wait$ is completely determined by $iE_{i+1}_rinput_delay$ and $iE_i_routput_wait$. The real world wait of $stage_i$ is caused by either or both of the following two facts: $stage_i$ is producing at a faster than 'normal speed' and/or $stage_{i+1}$ is producing at a slower than 'normal speed'. The 'normal speed' used in this analysis refers to the ideal world production rate. In this situation, if the ideal

world measurement is used as a reference, then element$_i$ causes E_i_output_wait by iE_i_routput_wait and element$_{i+1}$ causes E_i_output_wait by iE_{i+1}_rinput_delay.

The following assumption states that stage$_i$ causes a real world wait between stage$_i$ and its neighbouring stage by the amount of wait it would otherwise experience if in an ideal surrounding environment which is exactly the same as that provided by the 'ideal world'.

assumption: Element$_i$ always causes E_i_output_wait and E_i_input_wait by iE_i_routput_wait and iE_i_rinput_wait, respectively.

Under the assumption, element$_{i+1}$ in general, causes E_i_output_wait$_j$ by iE_{i+1}_late$_{kj}$, where k is the current cycle of element$_{i+1}$. According to the definition of iE_{i+1}_late in equation 6.2, every iE_{i+1}_input_delay has an effect on all subsequent E_i_output_wait. This effect is lessened by the increase of iE_i_total_output_delay. Therefore, the quantified amount by which element$_{i+1}$ causes E_i_output_wait is a relative measurement which is dependent on the cumulative ideal world performances of element$_i$ and element$_{i+1}$.

According to equation 5.62, the updating of E_i_shifted is based on the duration of waits in both the real world and the ideal world of element$_i$ of the current cycle. The cause of an ideal world wait on an element is the element itself according to the assumption. Thus, E_i_shifted has a recursive relationship with the real world waits. The causes for an element to be 'shifted' are then already determined if the causes of the real world wait have been determined. Therefore, the assumption contained in this section

alone is sufficient to determine the causes of a real world wait between two elements.

The causes between the past and the present of a real world wait can be distinguished by noticing that $iE_i_total_output_delay_{j-1}$ and $iE_{i+1}_total_input_delay_{k-1}$ are cumulative values of the ideal world measurements. Together with $E_i_shifted_{j-1}$ and $E_{i+1}_shifted_{k-1}$ they represent the effect of the past up to the previous cycle. The effect of these four parameters on the current $E_i_output_wait$ is shown in equations 6.1 and 6.2.

6.2.2 Real World Input Wait Causes

The same analysis can be applied to the real world input wait calculation using equation 5.28, which has been reproduced here:

$$E_i_input_wait_j = \left(iE_{i-1}_late_{hj} - E_{i-1}_shifted_{h-1} \right) + \left(iE_i_input_wait_j + E_i_shifted_{j-1} \right) \qquad 6.3$$

where

$$iE_{i-1}_late_{hj} = iE_{i-1}_output_delay_h - iE_i_total_input_delay_{j-1} + iE_{i-1}_total_output_delay_{h-1} \qquad 6.4$$

Under the same assumption, $element_i$ causes $E_i_input_wait_j$ by $iE_i_input_wait_j$ and $element_{i-1}$ causes $E_i_input_wait_j$ by $iE_{i-1}_late_{hj}$. $iE_{i-1}_late_{hj}$ is a relative measurement dependent upon the cumulative ideal world performances of $element_{i-1}$ and $element_i$.

The causes of the real world waits between two neighbouring elements are quantified by the information given by the 'frame work', and the assumption contained in this section. As stated earlier, $E_i_shifted$ has a recursive relationship with the real world waits. However, $E_i_shifted$ is given as a net result and the information about the

cause and effect of this result has not been explicitly expressed. A schema based on the current assumption of quantifying the causes of a real world wait between two elements will be presented in the next section which provides further detailed information about the causes of E_i_shifted. This will also make possible the determination of the causes of the real world waits by all of the elements in the system.

6.3 CAUSES AND EFFECTS OF REAL WORLD WAITS

This section details the causes of the real world waits by each element and their effects on the subsequent real world waits of other elements in a system. A schema for this purpose has been derived. Define:

$$x = \sum_{i=1}^{n} x\hat{}_i \qquad 6.5$$

where x is a scalar and $x\hat{}$ is a column matrix which has n rows. n is the number of elements in the system. $x\hat{}_i$ is the item on the i^{th} row of $x\hat{}$.

E_i_shifted is the effect of any real world wait experienced by $element_i$ according to equation 5.62, and is the cause of other subsequent real world waits of $element_{i+1}$ and $element_{i-1}$ according to equations 6.1 and 6.3. E_i_shifted$\hat{}$ is the result of the application of the format described by equation 6.5 to E_i_shifted. The objective is to distinguish the effect of the l^{th} element (l=1..n) by recording it in the l^{th} row of E_i_shifted$\hat{}$ (i=1..n), which is also the cause of the l^{th} element from the position of $element_i$ to the subsequent real world waits.

Using the schema and the assumption used for quantifying the causes of a real world wait between two neighbouring elements, the real world output wait expression of

$element_i$ is then rewritten as is shown in equation 6.6 for a real world output wait. $Element_i$ causes $E_i_output_wait$ by $iE_i_routput_wait$. Therefore, $iE_i_routput_wait$ belongs to the i^{th} row. $Element_{i+1}$ causes $E_i_output_wait$ by iE_{i+1}_late. Therefore, iE_{i+1}_late is put in row i+1. $E_i_shifted\hat{}$ and $E_{i+1}_shifted\hat{}$ are initially defined to be zero matrices, since the initial values of $E_i_shifted$ and $E_{i+1}_shifted$ are zero.

$$E_i_output_wait_j\hat{} = \begin{bmatrix} 0 \\ : \\ 0 \\ iE_i_routput_wait_j \\ iE_{i+1}_late_{kj} \\ 0 \\ : \\ 0 \end{bmatrix} + E_i_shifted_{j-1}\hat{} - E_{i+1}_shifted_{k-1}\hat{} \qquad 6.6$$

In applying this schema to the real world input wait expression of $element_i$, it is rewritten as is shown in equation 6.7 for a positive real world input wait. $iE_{i-1}_late_{hj}$ and $iE_i_rinput_wait_j$ are in row i-1 and i respectively.

$$E_i_input_wait_j\hat{} = \begin{bmatrix} 0 \\ : \\ 0 \\ iE_{i-1}_late_{hj} \\ iE_i_rinput_wait_j \\ 0 \\ : \\ 0 \end{bmatrix} - E_{i-1}_shifted_{h-1}\hat{} + E_i_shifted_{j-1}\hat{} \qquad 6.7$$

The update of $E_i_shifted\hat{}$ now becomes:

$$E_i_shifted_j\hat{} = E_i_shifted_{j-1}\hat{} + \max(iE_i_output_wait_j\hat{}, iE_i_input_wait_j\hat{}) - \max(E_i_output_wait_j\hat{}, E_i_input_wait_j\hat{}) \quad \textbf{6.8}$$

where

$$iE_i_output_wait_j\hat{} = \begin{bmatrix} 0 \\ : \\ 0 \\ iE_i_output_wait_j \\ 0 \\ : \\ 0 \end{bmatrix} \quad \textbf{6.9}$$

if $iE_i_output_wait_j$ is positive. $iE_i_output_wait_j$ is in row i, and

$$iE_i_input_wait_j\hat{} = \begin{bmatrix} 0 \\ : \\ 0 \\ iE_i_input_wait_j \\ 0 \\ : \\ 0 \end{bmatrix} \quad \textbf{6.10}$$

if $iE_i_input_wait_j$ is positive. $iE_i_input_wait_j$ is in row i. Any negative ideal world or real world wait has no effect on the 'shifted' parameter. The row index of a variable in this schema is the same as the element index, so that E_i_x is in row i, except for the shifted^ matrices.

Expanding on a positive propagated output wait using the same schema, equation 5.52, which defines the length of a propagated output wait of $stage_i$ caused by $stage_L$ becomes:

$$E_i_propagated_output_wait_j\hat{} = \begin{bmatrix} iE_i_routput_wait_j \\ iE_{i+1}_routput_wait_k + iE_{i+1}_late_{kj} \\ iE_{i+2}_routput_wait_l + iE_{i+2}_late_{lk} \\ \vdots \\ iE_{L-1}_routput_wait_n + iE_{L-1}_late_{nm} \\ iE_L_late_{pn} \end{bmatrix} + E_i_shifted_{j-1}\hat{} - E_L_shifted_{p-1}\hat{} \qquad 6.11$$

This can be verified by substituting equation 6.6 into equation 5.52 successively.

The schema introduced in this section is illustrated in the following example. For simplicity's sake, it will be supposed that the system has only two stages and therefore, two elements. The storage in between the stages is initially full. This satisfies the pre-condition of a real world output wait for the first stage. Suppose that after the first stage finishes the first cycle, the ideal world measurements are $iE_1_output_wait_1 = 10$ and the others are zero. Further suppose that the ideal world measurements of the second stage after it finished the first cycle are $iE_2_input_delay_1 = 7$ and the others are zero. According to equation 6.6:

$$E_1_output_wait_1\hat{} = \begin{bmatrix} 10 \\ 7 \end{bmatrix} \qquad 6.12$$

then $E_1_output_wait_1 = 17$ according to equation 6.5. $E_1_shifted_1\hat{}$ is updated according to equation 6.8:

$$E_1_shifted_1\hat{} = \begin{bmatrix} 10 \\ 0 \end{bmatrix} - \begin{bmatrix} 10 \\ 7 \end{bmatrix} = \begin{bmatrix} 0 \\ -7 \end{bmatrix} \qquad 6.13$$

and $E_1_shifted_1 = -7$. This result shows that the current negative amount of E_1_shifted is

completely caused by element$_2$.

Further suppose that at cycle j of element$_1$ and cycle k of element$_2$, the storage in between the stages is empty, which satisfies the pre-condition for the real world input wait. Suppose E_1_shifted and E_2_shifted have not changed since the first cycle. If the ideal world measurements for element$_1$ after cycle j are iE_1_output_delay$_j$=8, iE_1_total_output_delay$_{j-1}$=11 and the others are zero and if the ideal world measurements of element$_2$ after cycle k are iE_2_input_wait$_k$=6, iE_2_total_input_delay$_{k-1}$=7 and the others are zero, then according to equation 6.7:

$$E_2_input_wait_k\hat{} = \begin{bmatrix} 12 \\ 6 \end{bmatrix} - \begin{bmatrix} 0 \\ -7 \end{bmatrix} = \begin{bmatrix} 12 \\ 13 \end{bmatrix} \qquad \textbf{6.14}$$

then E_2_input_wait$_k$=25. E_2_shifted$_k$^ is then updated as follows:

$$E_2_shifted_k\hat{} = \begin{bmatrix} 0 \\ 6 \end{bmatrix} - \begin{bmatrix} 12 \\ 13 \end{bmatrix} = \begin{bmatrix} -12 \\ -7 \end{bmatrix} \qquad \textbf{6.15}$$

then E_2_shifted$_k$=-19. This result distinguishes between the causes of the current amount of E_2_shifted$_k$ by the two elements. The cause by element$_2$ in the current E_2_shifted$_k$ is due to its previous effect on element$_1$.

The measurements in this simplified example are rather hypothetical. Nevertheless, it demonstrates the application of the schema developed in this chapter. The same schema can be applied to systems with any number of stations with arbitrary processing times, since its correctness has been shown in chapter five. The schema quantifies the cause and effect relationships between the elements in a system and the result of the amount each element has 'shifted' can be traced to the net cause by each individual element in the system, each of which is recorded in a row of a column matrix.

6.3.1 System Monitoring

The analysis in this chapter should have satisfied the general requirements of on-line system monitoring which were stated in chapter three. The quantitative cause and effect relationship of production loss among the elements has been shown by the schema in this chapter. System production loss is reflected by the production loss of the output element in the system.

The 'blocking', either direct or propagated, and the 'starvation' of an element are the sources of production loss. The schema first distinguishes the causes of any 'blocking' and 'starvation' among the elements in the system. The effect of these causes is then recorded individually in the 'shifted' parameter of an element using a column matrix. The amount an element is 'shifted' contributes directly to the production loss of the element according to the 'real world performance' section contained in chapter five. Therefore, the 'shifted' matrix of an element at any instant reviews the effect of production loss caused individually by each of the other elements in the system at that instant. Figure 6.9 at the end of this chapter shows the system monitoring procedure being incorporated into the real world parameter calculation procedure. The real world parameter calculation procedure is executed for $stage_i$ ($i = 1..n$) at each of its end of cycle times. The block diagram of the real world parameter calculation procedure was first shown in figure 5.11.

Secondly, the analysis can be performed on a real time basis since all of the calculations involved are deterministic. Therefore, it is believed that the development of the analysis up to this point have satisfied the general requirements of on-line system monitoring.

6.4 EMULATION EXAMPLES

This section further shows the application of the schema by using emulation examples. Emulation has been defined in the nomenclature as the process of using computer programs to artificially generate all of the necessary parameters for this study. The logical relationships among these generated parameters are the same as those of the real parameters in a real production system, given the assumptions of this study. Alternatively, these parameters can be gathered directly from a real production system.

The emulated system has three elements. The cycle times in this case are generated randomly following a uniform distribution with a mean cycle time chosen to be 60 time units. Each inter-stage storage has two units and is half full. The storage space in the previous example is of no concern since the ideal world measurements in that example were hypothetically supposed.

In figure 6.1, iE_1_total_output_delay and iE_2_total_input_delay are plotted against the number of jobs produced. The two curves increase monotonically since they are the cumulative values of the positive ideal world delays. It happens that iE_2_total_input_delay is always greater than iE_1_total_output_delay during this time period when 5,000 jobs are produced.

In figure 6.2, iE_2_total_output_delay and iE_3_total_input_delay are plotted against the number of jobs produced. The two curves again increase monotonically and it happens that the iE_2_total_output_delay is always greater than the iE_3_total_input_delay in that time period.

In figure 6.3, the values of the three rows of E_3_shifted^ are plotted against the number of jobs produced by $stage_3$. The values of E_3_shifted^ are of interest since the

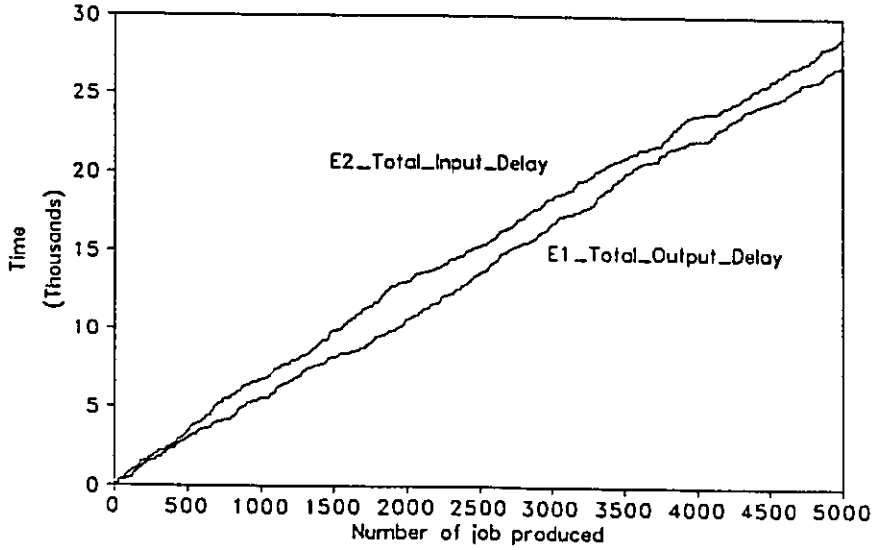


Figure 6.1 Cumulative ideal world delays vs. number of jobs

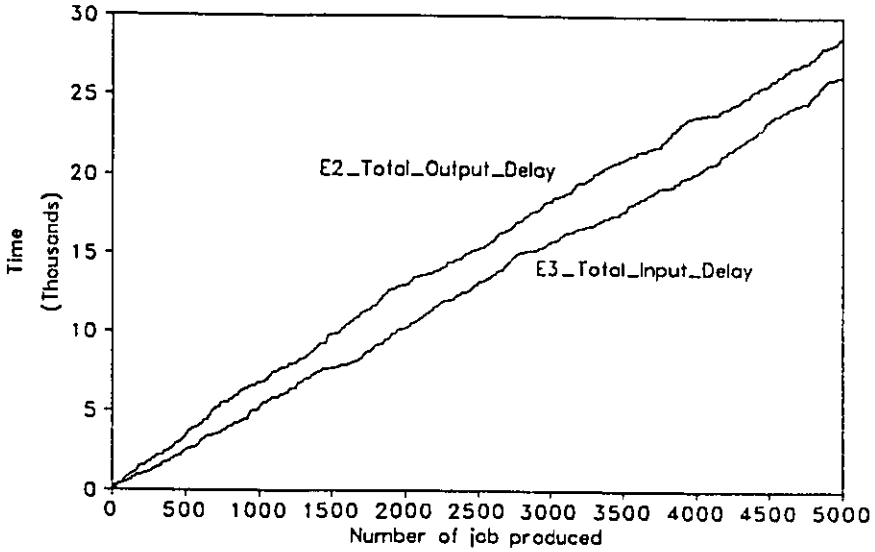


Figure 6.2 Cumulative ideal world delays vs. number of jobs

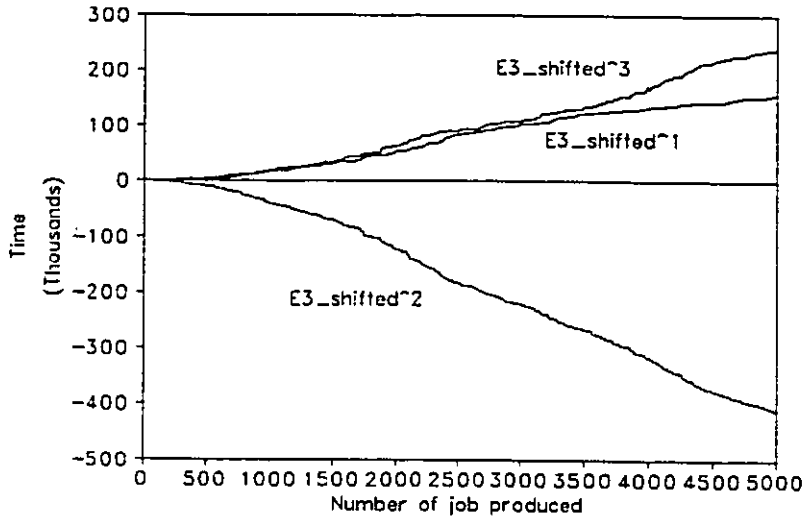


Figure 6.3 Itemized amount of element$_3$ 'shifted' vs. number of jobs

output of stage$_3$ is the output of the system in a three stage system and E_3_shifted^ contributes directly to the system's performance. In that figure, E_3_shifted^$_1$ and E_3_shifted^$_3$ are increasing and E_3_shifted^$_2$ is decreasing. This is consistent with the cumulative ideal world delays which were shown in figures 6.1 and 6.2.

According to the assumption contained in this chapter, iE_i_late directly contributes to the real world waits of the other elements. The difference between E_i_total_output_delay and E_{i+1}_total_input_delay contributes to the value of iE_i_late. In this case, it is very likely that iE_2_late is greater than iE_1_late for real world waits between stage$_1$ and stage$_2$ and iE_2_late is greater than iE_3_late for real world waits between stage$_2$ and stage$_3$. Therefore, element$_2$ contributes to the negative shifting of element$_3$. In other words, it contributes to the system's production loss. Element$_1$ and

$element_3$ contribute positively to the shifting of the system's output relative to $element_2$, which is a relative production gain in the given situation.

The emulated production time (E_i_Sct) for this example is 341,416.88 time units. Using the efficiency definition of equation 5.60, the efficiency of the system for producing 5,000 jobs is 87.87 percent.

The following is a second example which complements the first one. The initial settings for this example are exactly the same as those for the first one, except that different sets of random processing cycle times are used. The same three diagrams are plotted.

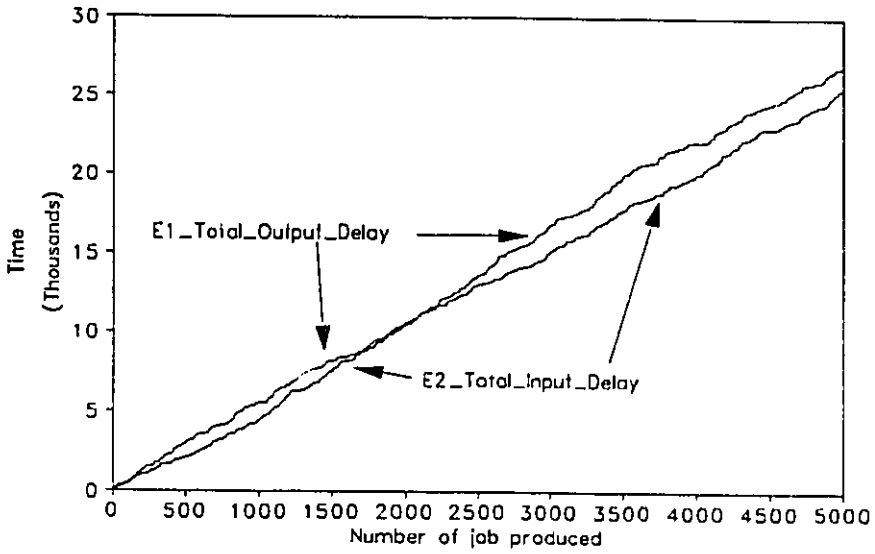


Figure 6.4 Cumulative ideal world delays vs. number of jobs

In this case, iE_2_total_input_delay is less than iE_1_total_output_delay, and iE_2_total_output_delay is less than iE_3_total_input_delay. These results are opposite to those of the first example. The results shown in figure 6.6 agree with these inequalities.

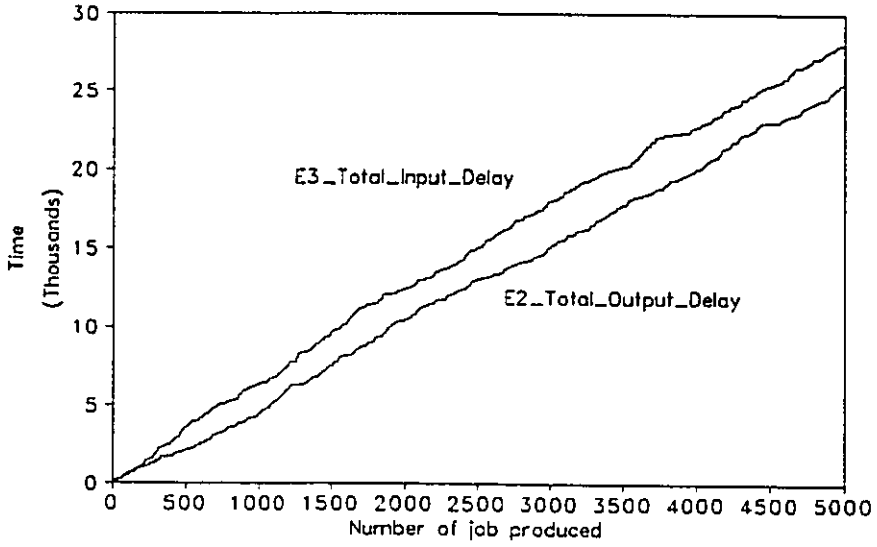


Figure 6.5 Cumulative ideal world delays vs. number of jobs

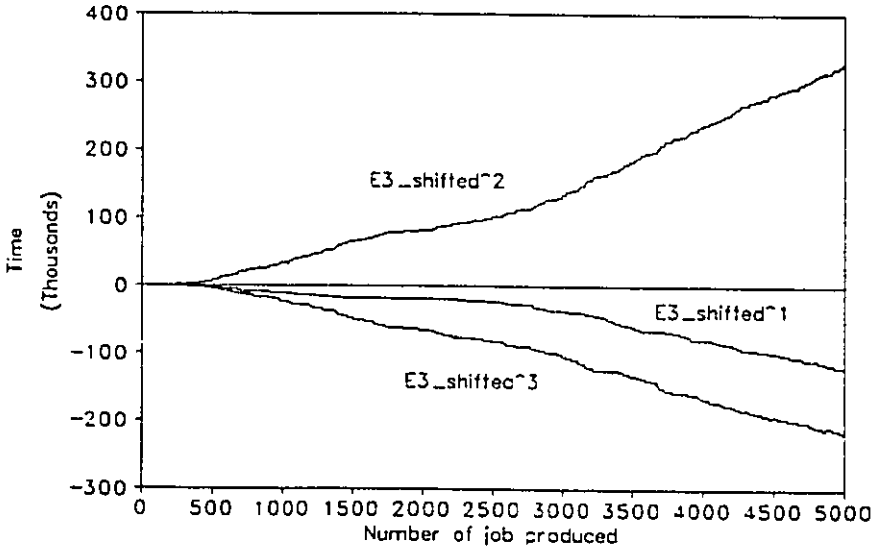


Figure 6.6 Itemized amount of element$_3$ 'shifted' vs. number of jobs

The emulated production time for this example is 339,230.24 time units and the efficiency of the system for producing 5,000 jobs is 88.44 percent.

The means and variances of all of the processing cycle times used in these examples are close to being equal, and each inter-stage storage space is equal to the other. Each storage space is divided up equally when setting up the elements in the abstract environment, and the initial number of jobs in all of the elements is the same. It is conjectured that the difference between the results of the two examples is due to the sequencing of the processing cycle times. For example, a series of long cycle times followed by a series of short cycle times can result in a higher total ideal world delays than if the long and short cycle times are arranged alternatively.

6.4.1 Summary of Emulation Examples

The importance of these results, besides the fact that they can be applied to the defined system monitoring problem, is that they show the direct relationships between the independent measurements and the system performance, which is that the resulting system performance shown in figures 6.3 and 6.6 are dictated by the independent measurement results shown in figures 6.1, 6.2 and 6.4, 6.5 respectively, and the qualitative reasoning is straightforward. More importantly, this implies that problems concerning the system as a whole such as the controlling or designing of certain system parameters can be achieved by manipulating the independent measurement of each element individually according to an overall criteria, which is set using the relationships between the independent measurements and the real world parameters. In other words, a problem concerning a system can be decomposed into smaller problems each of which

concerns only one element. This introduces a new approach to problem solving with respect to the underlying system. The success of this approach will be demonstrated in the examples of system production control and optimal storage allocation problems in the next chapter.

The results of the two examples, although limited in number, have been randomly chosen. The remainder of this section shows that the results of the two examples based on the current analysis agrees with existing phenomena concerning systems with inter-stage storages.

The first phenomenon is known as the "bowl phenomenon". For a three stage system, the "bowl phenomenon" (Hillier & Boling 66) conjectured that if a lower average operation time is assigned to the stage in the middle than to the stages at the two ends, a slight improvement in the efficiency over that of a perfectly balanced system will result. (Gershwin 87) stated that for maximal production rate, more storage space should be allocated to the middle of a system than to the rest of the system. The "bowl phenomenon" in both instances seems to agree that the bottle neck area is at the middle of a system, and a higher performance in this area than for the rest of the system will result in a slightly higher system efficiency than that found in a balanced system. This phenomenon manifests itself in a slightly different form in the two emulation examples contained in this chapter, since the average cycle times and storage allocations in all three stages are equal. According to the results of the analysis, figure 6.3 of the first example shows that the middle element has a negative effect on the system's output relative to the other two elements. In figure 6.6 of the second example, the middle element has a positive effect on the system output relative to the other two elements and the system

efficiency is more than 0.6 percent higher in the second example than in the first example.

The second phenomenon concerns the reversibility property. (Hillier & Boling 66) stated that the reversibility property for a three stage system is that the system production rate function is symmetrical with respect to the average operation times of stage 1 and stage 3. In other words, stages 1 and 3 are interchangeable in terms of the effect their average work times have on the system's production rate.

An example of a three-stage system which shows the reversibility property of the current analysis is contained in appendix H. The inter-stage storages of the system are all arbitrarily set to one and are originally empty. Figure H.1 in appendix H shows $E_3_shifted\hat{}_i$ (i=1..3), when the mean cycle times of the three stages are as shown in table H.1. Figure H.2 is generated after inter-changing the cycle times of stage 1 with stage 3. It can be noticed that figures H.1 and H.2 are almost identical except that $E_3_shifted\hat{}_1$ and $E_3_shifted\hat{}_3$ are also inter-changed in the two diagrams. Therefore, the net effects $E_3_shifted\hat{}_i$ (i=1..3) as shown by the two diagrams have similar effects on the system. This example shows that the result of the current analysis, namely that $E_3_shifted\hat{}_i$ (i=1..3) for a three-stage system, agrees with the reversibility phenomenon.

6.5 RELATIONSHIPS BETWEEN REAL WORLD WAITS AND ELEMENT SHIFTING

In this section, the change of $E_i_shifted$ after each real world wait is examined for the purpose of further understanding the cause and effect of production loss in the system. These changes are expressed in the matrix form of the schema defined in this

chapter. $\Delta E_i_shifted_j$ is defined to be the change of $E_i_shifted$ after finishing cycle j. Using equation 5.62:

$$\Delta E_i_shifted_j = \max(iE_i_output_wait_j, iE_i_input_wait_j) - \max(E_i_output_wait_j, E_i_input_wait_j, 0) \quad \textbf{6.16}$$

and if $E_i_output_wait_j$ and/or $E_i_input_wait_j$ are positive, then using equations 6.9 and 6.10:

$$\Delta iE_i_shifted_j\hat{} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \max(iE_i_output_wait_j, iE_i_input_wait_j) \\ 0 \\ \vdots \\ 0 \end{bmatrix} - \max(E_i_output_wait_j\hat{}, E_i_input_wait_j\hat{}) \quad \textbf{6.17}$$

There are several different cases where $E_i_shifted_j$ can change. These cases are combinations of the real world and ideal world, input and output waits. The following sections show these cases individually.

case 1: (real world output wait > 0) and (real world output wait > real world input wait). Using equations 6.6 and 6.17:

$$\Delta E_i_shifted_j\hat{} = \begin{bmatrix} 0 \\ : \\ 0 \\ iE_i_\alpha_j \\ -iE_{i+1}_late_{kj} \\ 0 \\ : \\ 0 \end{bmatrix} - E_i_shifted_{j-1}\hat{} + E_{i+1}_shifted_{k-1}\hat{} \qquad \textbf{6.18}$$

where

$$iE_i_\alpha_j = \max(-iE_i_output_wait_j, iE_i_input_wait_j) - iE_i_output_wait_j \qquad \textbf{6.19}$$

and since

$$iE_i_output_wait_j = pos(\ iE_i_output_wait_j\ ) \qquad \textbf{6.20}$$

then

$$iE_i_\alpha_j \geq 0 \qquad \textbf{6.21}$$

case 2: (real world input wait > 0) and (real world input wait > real world output wait). Using equations 6.7 and 6.17:

$$\Delta E_{i_}\text{shifted}_j\hat{} = \begin{bmatrix} 0 \\ : \\ 0 \\ -iE_{i-1_}\text{late}_{hj} \\ iE_{i_}\beta_j \\ 0 \\ : \\ 0 \end{bmatrix} + E_{i-1_}\text{shifted}_{h-1}\hat{} - E_{i_}\text{shifted}_{j-1}\hat{} \qquad 6.22$$

where

$$iE_{i_}\beta_j = \max(\, iE_{i_}\text{output_wait}_j,\ iE_{i_}\text{input_wait}_j \,) - iE_{i_}\text{rinput_wait}_j \qquad 6.23$$

and since

$$iE_{i_}\text{input_wait}_j = \text{pos}(\, iE_{i_}\text{rinput_wait}_j \,) \qquad 6.24$$

then

$$iE_{i_}\beta_j \geq 0 \qquad 6.25$$

case 3 and 4: Both the real world output wait and real world input wait are zero, and at least one of the ideal world output wait and ideal world input wait is non-zero. Using equation 6.17, in any of these two cases:

$$\Delta E_i_shifted_j\hat{} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \max(\, iE_i_output_wait_j,\ iE_i_input_wait_j \,) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad 6.26$$

where row i is the only non-zero row.

case 5: (real world propagated output wait > real world input wait). Using equations 6.11 and 6.17:

$$\Delta E_i_shifted_j\hat{} = \begin{bmatrix} iE_i_\alpha_j \\ -(\, iE_{i+1}routput_wait_k + iE_{i+1}_late_{kj} \,) \\ -(\, iE_{i+2}routput_wait_l + iE_{i+2}_late_{lk} \,) \\ \vdots \\ -(\, iE_{L-1}routput_wait_n + iE_{L-1}_late_{nm}) \\ -iE_L_late_{pn} \end{bmatrix} - E_i_shifted_{j-1}\hat{} + E_L_shifted_{p-1}\hat{} \qquad 6.27$$

$iE_i_\alpha_j$ is the same as that found in equation 6.18.

6.5.1 Summary

In summary, the relationships between real world waits and the updated $E_i_shifted$ of an element has been expressed for the five cases using:

$$E_i_shifted_j\hat{} = \Delta E_i_shifted_j\hat{} + E_i_shifted_{j-1}\hat{} \qquad 6.28$$

Therefore, using equation 6.18 for case 1:

$$E_i_shifted_j\hat{} = \begin{bmatrix} 0 \\ : \\ 0 \\ iE_i_\alpha_j \\ -iE_{i+1}_late_{kj} \\ 0 \\ : \\ 0 \end{bmatrix} + E_{i+1}_shifted_{k-1}\hat{} \qquad 6.29$$

Using equation 6.22 for case 2:

$$E_i_shifted_j\hat{} = \begin{bmatrix} 0 \\ : \\ 0 \\ -iE_{i-1}_late_{hj} \\ iE_i_\beta_j \\ 0 \\ : \\ 0 \end{bmatrix} + E_{i-1}_shifted_{h-1}\hat{} \qquad 6.30$$

and rewritten for $element_i$ and $element_{i+1}$:

$$E_{i+1}_shifted_k\hat{} = \begin{bmatrix} 0 \\ : \\ 0 \\ -iE_i_late_{jk} \\ iE_{i+1}_\beta_k \\ 0 \\ : \\ 0 \end{bmatrix} + E_i_shifted_{j-1}\hat{} \qquad 6.31$$

Using equation 6.26 for cases 3 and 4:

$$E_i_shifted_j\hat{} = \begin{bmatrix} 0 \\ : \\ 0 \\ \max(\ iE_i_output_wait_j,\ iE_i_input_wait_j\ ) \\ 0 \\ : \\ 0 \end{bmatrix} + E_i_shifted_{j-1}\hat{} \qquad 6.32$$

and rewritten for $element_{i+1}$:

$$E_{i+1}_shifted_k\hat{} = \begin{bmatrix} 0 \\ : \\ 0 \\ \max(\ iE_{i+1}_output_wait_k,\ iE_{i+1}_input_wait_k\ ) \\ 0 \\ : \\ 0 \end{bmatrix} + E_{i+1}_shifted_{k-1}\hat{} \qquad 6.33$$

Using equation 6.27 for case 5:

$$E_i_shifted_j\hat{} = \begin{bmatrix} iE_i_\alpha_j \\ -(\ iE_{i+1}_routput_wait_k + iE_{i+1}_late_{kj}\ ) \\ -(\ iE_{i+2}_routput_wait_l + iE_{i+2}_late_{lk}\ ) \\ \vdots \\ -(\ iE_{L-1}_routput_wait_n + iE_{L-1}_late_{nm}) \\ -iE_L_late_{pn} \end{bmatrix} + E_L_shifted_{p-1} \qquad 6.34$$

6.6 SOFTWARE IMPLEMENTATION

The C++ language was chosen for exploratory programming in this study. For convenience's sake, each element of the system representation was implemented as an object. Each object of the implementation inherits from the two classes, the ideal world class and the real world class, as shown in figure 6.7.

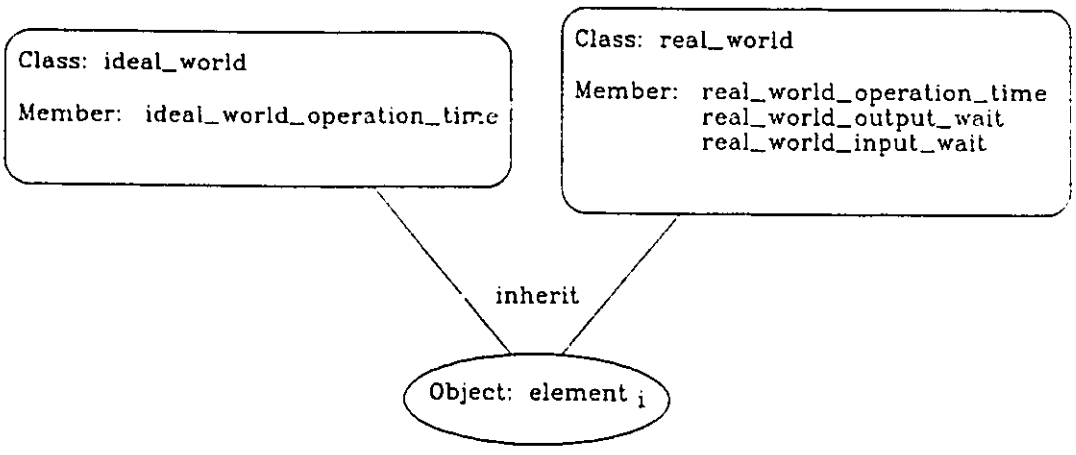


Figure 6.7 The classes of the object of an element

For serial configuration, which is the case that has been considered up to this

point, there are two types of sub-classes of the class 'ideal world': the 'input' sub-class and the 'output' sub-class as shown in figure 6.8.

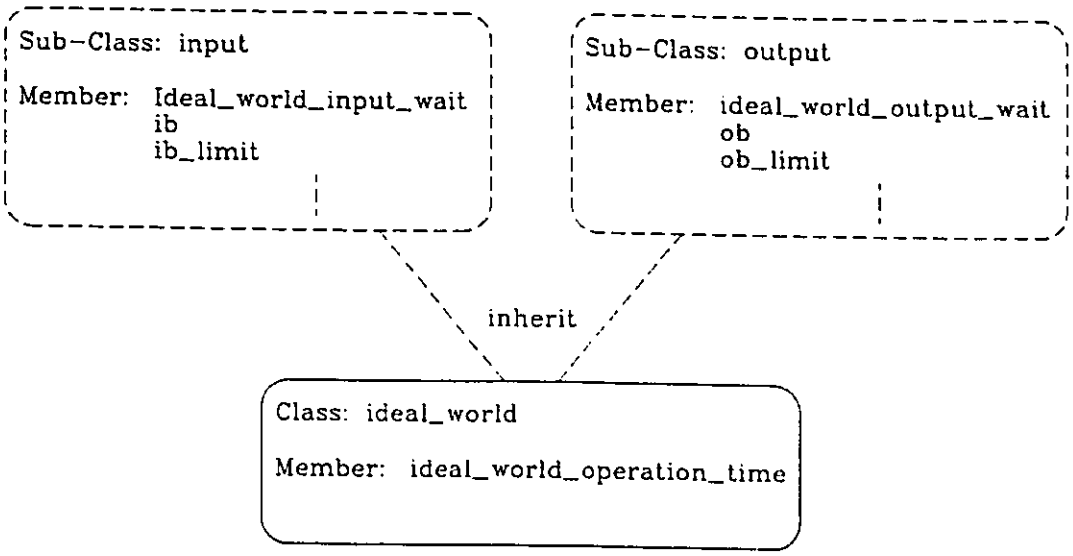


Figure 6.8 The sub-classes of the ideal world class

6.7 CONCLUSION

In this chapter, the causes and effects of production loss have been defined and a schema has been derived to quantify these relationships. Production loss is due to 'blocking and starvation' or equivalently, the real world waits of any element in the system. An assumption has been established in section 6.2 to quantify the causes of a real world wait by any two neighbouring elements. The effect of production loss of an element in this analysis manifests itself as changes in E_i_shifted of the element. E_n_shifted of the output element shows the effect production loss has on the system. A schema has been established in this chapter based on this assumption so that the effect

of production loss can be quantitatively traced back to the causes by each element in the system by recording $E_{i_}$shifted in a column matrix. The results of this schema are shown through examples which agree with the known properties of systems which have three stages.

The results shown in figure 6.3 and 6.6 'explains' quantitatively the current system performance by tracing the causes to each stage. These results can be produced on a real-time basis, thus satisfying the system monitoring requirement. The results for system monitoring is in turn 'explained' by the independent measurement results shown in figures 6.1, 6.2 and 6.4, 6.5. This implies that problems concerning the system can be decomposed into smaller problems, each of which concerns only one element. Thus a new approach to solving problems concerning the system has been introduced. The success of this approach will be demonstrated in the examples of system production control and optimal storage allocation problems in the next chapter.

The updated $E_{i_}$shifted has been expressed in terms of events which happen in the system. These events are combinations of real world and ideal world, output and input waits. These expressions are useful for analysing the current situation of the system and provide insights to the system so that inferences about the system can be drawn.

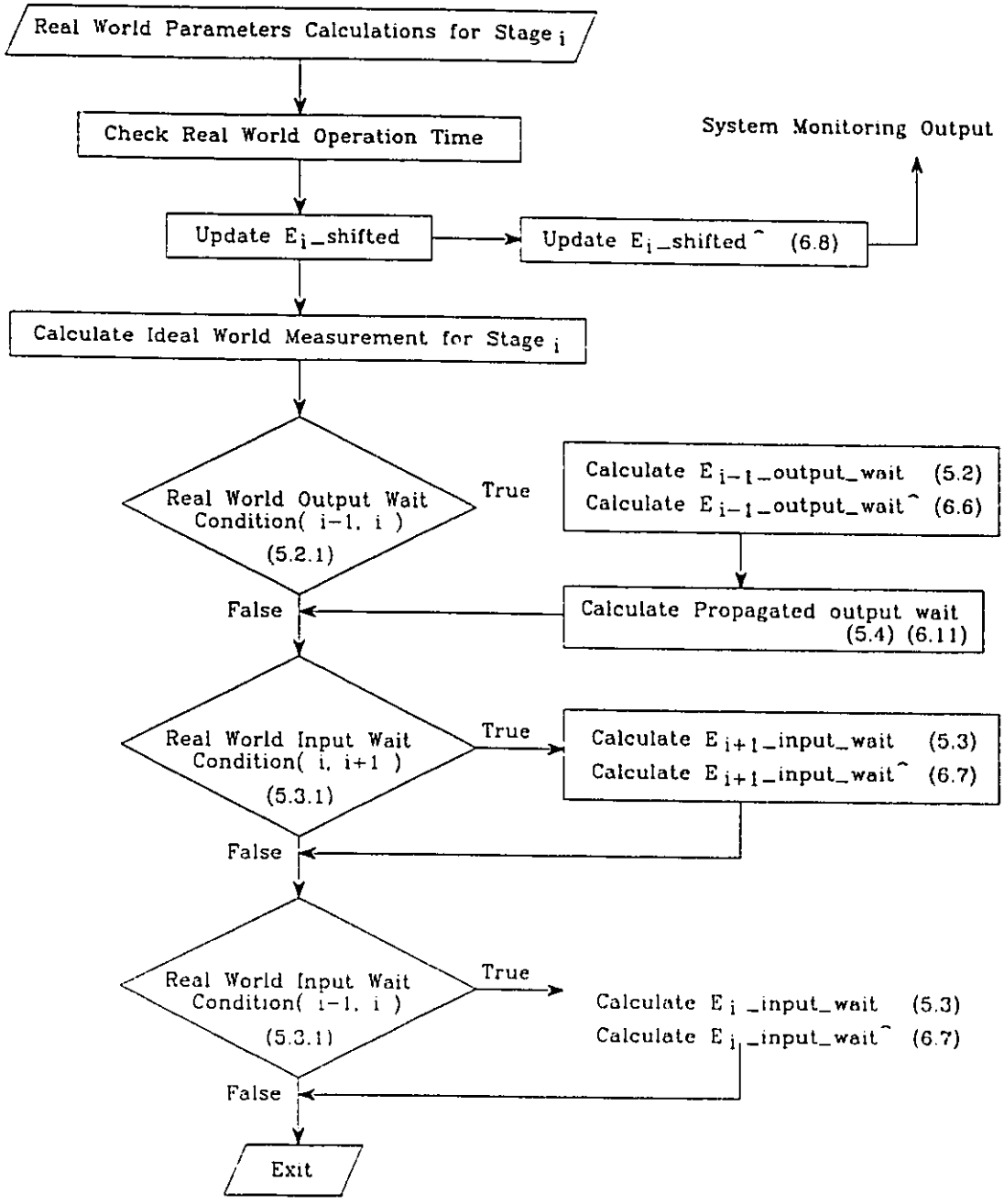


Figure 6.9 Block diagram for the real world parameter calculation and system monitoring procedures

CHAPTER 7

APPLICATION EXAMPLES

7.1 INTRODUCTION

In this chapter, practical examples illustrating the methodology and the analysis contained in chapter six are presented. The analysis in chapter six has quantified by how much each of the elements have caused the real world waits and how much this causation has effected the system output based on the instantaneous cause and effect relationships. Thus, the basic requirement of system monitoring has been satisfied. In addition, it is obvious that inferences about system behaviour, which can be drawn from the production loss cause and effect relationships of the current approach, can lead to more interesting applications. The first example presented in this chapter is about production control and is motivated by and related to system monitoring. The second example about storage allocation is a simple one which shows that the methodology and the analysis in chapter six can provide insights about system behaviour which can be used in system design.

7.2 EXAMPLE I: THE PRODUCTION CONTROL EXAMPLE

In this example, a 'maximum cycle time algorithm' will be developed. This algorithm calculates the maximum cycle time for each element for each cycle of an emulated or real production system. It will be shown that if the cycle time of each stage

at each cycle could be limited to the calculated maximum given by the algorithm, the production time needed to produce a fixed amount of jobs will match the expected time.

The algorithm is constrained by the fact that the calculated maximum cycle time for an element should not be less than the designed mean cycle time minus the output transfer time included in the element. Furthermore, the algorithm treats each element equally. The expected production time is defined as the designed mean cycle time of any stage multiplied by the expected amount of system output. The designed mean cycle times of all of the stages are assumed to be equal in a balanced serial system, as stated in chapter one. Therefore, the result of applying the algorithm is that the system is controlled so that one job will be produced at each time interval equal to the mean cycle time.

In this example, the criterion for achieving the goal of matching the expected production time with the resulting production time has been setup by analysing the cause and effect relationships of production loss which are contained in chapter six. An algorithm based on the criterion is developed. The algorithm monitors the processing cycle time of each stage at each cycle in the system and depending on the state of the system, calculates a maximum cycle time. If each stage abides by the maximum cycle time for every cycle, the system should have a production time close to the expected time.

The 'maximum cycle time' information can be helpful to on-line system monitoring and even to job scheduling, such as the mixing of different models in a production line. For each of the following examples, the changes in the efficiency as defined in chapter five and the mean processing times, after the maximum cycle time

algorithm has been applied, are compared to their original values.

7.2.1 The Maximum Cycle Time Algorithm

The real world operation time which is measured at the starting point of a processing cycle has been defined in equation 5.59, and has been reproduced here for the output element at the j^{th} cycle:

$$E_n_Sct_j = (iE_n_j_{out} - iE_n_ob_j + E_n_N_i) \times mct + iE_n_total_output_delay_{j-1} - E_n_shifted_{j-1} \quad 7.1$$

When the production time is large, the production loss is dominated by $iE_n_total_output_delay_j$ and $E_n_shifted_{j-1}$. The magnitude of $(iE_n_j_{out} - iE_n_ob) \times mct$ is relatively small, since the output storage size is finite and is often quite limited. Therefore, the objective of the algorithm is to minimize $(iE_n_total_output_delay_{j-1} - E_n_shifted_{j-1})$, so that $E_n_Sct_j$ should be as close to $(E_n_N_i \times mct)$ as possible. The algorithm is simply a set of rules derived from qualitative reasoning which uses the results contained in chapter six. Its effect is shown by using various random examples rather than by mathematical proof.

After expressing the two terms in column form, $(iE_n_total_output_delay_{j-1}\hat{} - E_n_shifted_{j-1}\hat{})$ can be minimized in many different ways, as long as the sum of all of the rows of the two matrices is minimized. For simplicity's sake, it is proposed that $E_n_shifted$ be minimized by minimizing the magnitude of all $E_n_shifted\hat{}_c$ equally, where $c=1..n-1$. This way, all of the stages are treated equally except for the output stage. For the output stage, a positive $E_n_shifted\hat{}_n$ is needed to offset the value of $iE_n_total_output_delay$. It will be shown later that intuitively, the constraint of treating

every stage equally has not been violated.

7.2.1.1 The Preliminary Criterion

The preliminary criterion tries to prevent any $E_i_shifted^{\wedge}_c$ ($i=1..n$, $c=1..n$) from becoming negative, as the first step towards reaching the objective of the algorithm. By inspecting equations 6.29 to 6.33, with the knowledge that all $E_i_shifted^{\wedge}_c$ ($i=1..n$, $c=1..n$) are initially zero, the only possible causes for $E_i_shifted^{\wedge}_c$ being negative are iE_{i+1}_late and iE_{i-1}_late. The effect of propagated output waits are ignored for the moment and the reason for doing so will be justified later. Therefore, minimizing iE_{i+1}_late and iE_{i-1}_late will minimize the negative effect on $E_i_shifted^{\wedge}_c$. This can be satisfied by eliminating both the $iE_i_output_delay$ and $iE_i_input_delay$ for all i, according to the definitions of iE_{i+1}_late and iE_{i-1}_late. $iE_i_input_delay$ can be eliminated if the cycle times of a stage are bounded above. In calling the upper bound of the cycle time for element$_i$ at cycle j $E_i_max_{pi}_ct_j$, where the subscript $_{pi}$ stands for preliminary and input_delay, then according to chapter four:

$$E_i_max_{pi}_ct_j = (iE_i_ib_limit - iE_i_ib_{j-1} + 1) \times E_i_mct \qquad 7.2$$

$E_i_max_{pi}_ct_j$ has a minimum value of E_i_mct, since the maximum value of $iE_i_ib_{j-1}$ is $iE_i_ib_limit$. The cycle time upper bound for eliminating $iE_i_output_delay$ is:

$$E_i_max_{po}_ct_j = iE_i_ob_{j-1} \times E_i_mct + iE_i_\theta_{j-1} - iE_i_output_transfer_time \qquad 7.3$$

$E_i_max_{po}_ct_j$ has a minimum value of (E_i_mct - $iE_i_output_transfer_time$). $iE_i_\theta_{j-1}$ will be always zero if $iE_i_output_delay$ is always zero. Thus, the preliminary criterion is that $E_i_ct_j$ is to be bounded by both of these cycle time limits. The preliminary criterion

therefore is:

$$E_i_ct_j \leq E_i_max_p_ct_j = \min(E_i_max_{pi}_ct, E_i_max_{po}_ct) \qquad 7.4$$

The minimum value of the specified maximum cycle time satisfied the constrained minimum value, and it is close to E_i_mct, since iE_i_output_transfer_time is a fraction of the real output transfer time between $element_i$ and $element_{i+1}$, which should be designed to be much less than E_i_mct.

The preliminary criterion prevents all E_i_shifted^$_c$ (i=1..n, c=1..n) from becoming negative, including E_n_shifted^. Therefore it is the first step towards reaching the objective of the algorithm. However, if the preliminary criterion is satisfied by all of the stages, the result is that the output element will be positively shifted after a long enough operation time. Although a positive E_n_shifted^ is not as undesirable as a negative one, the result still does not meet the objective of the algorithm. The following criterion shows how E_i_shifted^$_c$ (i=1..n, c=1..n) can be brought closer to the desired value. For clarity's sake, it should be kept in mind that in the column matrices of the schema, the row index of a variable is the same as its element index, so that iE_i_α is in row i, except for E_i_shifted^ (i=1..n).

7.2.1.2 The Criterion

The criterion is based on the preliminary criterion and if satisfied by all of the stages in a system, the system production time should closely match the expected time for a certain amount of output. This criterion relaxes the preliminary criterion by allowing certain negative effects on E_i_shifted^$_c$ (i=1..n, c=1..n) when they are greater

than the desired values while preventing a negative effect on these values when they are close to the desired values.

The derivation of the criterion is strictly qualitative, since no knowledge is assumed about the sequence, the possibility of occurrences of events in the system and the values of any variable (except initial conditions). The derivation process concentrates on element$_i$ and element$_{i+1}$ with i=1..n-1, so that the criterion applies equally to each and every element in the system. The results from chapter six provides the basis for the reasoning, especially equations 6.29 to 6.33. Equation 6.31 is used instead of 6.30, since element$_{i+1}$ is used instead of element$_i$ for this situation. Equations 6.32 and 6.33 are for element$_i$ and element$_{i+1}$ respectively under the same situation.

7.2.1.2.1 Argument for The Criterion

It is assumed that the preliminary criterion is in effect and is observed by all of the stages. $E_i_shifted\hat{}_c$ (i=1..n, c=1..n) are initially zero. Firstly, it is examined how $E_i_shifted\hat{}_c$ can be controlled so that it remains close to zero even after a long operation period.

Consider $E_i_shifted\hat{}_i$ which is the i^{th} row of $E_i_shifted\hat{}$, and $E_{i+1}_shifted\hat{}_{i+1}$ which is the $i+1^{th}$ row of $E_{i+1}_shifted\hat{}$. If the situation described by equation 6.31 or 6.33 occurs, $E_{i+1}_shifted\hat{}_{i+1}$ can be incremented and may have a positive value. In addition, if the situation described by equation 6.29 or 6.32 occurs, $E_i_shifted\hat{}_i$ can also become positive. The continuation of this situation will further increase $E_i_shifted\hat{}_i$ and $E_{i+1}_shifted\hat{}_{i+1}$.

From equations 6.29 and 6.31, the relationships between $E_i_shifted\hat{}_i$ and

$E_{i+1}_shifted\hat{}_i$ have been extracted as follows:

$$E_{i+1}_shifted_k\hat{}_i = -iE_i_late_{jk} + E_i_shifted_{j-1}\hat{}_i \quad 7.5$$

$$E_i_shifted_j\hat{}_i = iE_i_\alpha_j + E_{i+1}_shifted_{k-1}\hat{}_i \quad 7.6$$

depending on which situation has occurred. The relationships between $E_{i+1}_shifted\hat{}_{i+1}$ and $E_i_shifted\hat{}_{i+1}$ have been extracted as follows:

$$E_i_shifted_j\hat{}_{i+1} = -iE_{i+1}_late_{kj} + E_{i+1}_shifted_{k-1}\hat{}_{i+1} \quad 7.7$$

$$E_{i+1}_shifted_k\hat{}_{i+1} = iE_{i+1}_\beta_k + E_i_shifted_{j-1}\hat{}_{i+1} \quad 7.8$$

Equations 7.5 and 7.6 for $E_i_shifted\hat{}_i$ are symmetrical to equations 7.7 and 7.8 for $E_{i+1}_shifted\hat{}_{i+1}$. Therefore, any argument based on equations 7.5 and 7.6 for $E_i_shifted\hat{}_i$ can be applied to equations 7.7 and 7.8 for $E_{i+1}_shifted\hat{}_{i+1}$, and vice versa.

Continuing with the present reasoning, if $E_{i+1}_shifted\hat{}_{i+1}$ is positive, then $E_i_shifted\hat{}_{i+1}$ will become positive if the situation described by equation 7.7 happens since iE_{i+1}_late should be zero under the preliminary criterion. A positive $E_i_shifted\hat{}_{i+1}$ can be controlled by allowing a positive iE_{i+1}_late value, according to equation 7.7. If the magnitude of iE_{i+1}_late is close or equal to $E_{i+1}_shifted\hat{}_{i+1}$, then $E_i_shifted\hat{}_{i+1}$ will be close or equal to zero. Then if the situation described by equation 7.8 occurs, $E_{i+1}_shifted\hat{}_{i+1}$ will be close or equal to iE_{i+1}_β, which stops $E_{i+1}_shifted\hat{}_{i+1}$ from becoming increasingly larger.

A positive $iE_{i+1}_late_{kj}$ can be created by allowing a maximum cycle time which is greater than the one specified by the preliminary criterion, and is less than the value that would make iE_{i+1}_late greater than $E_{i+1}_shifted\hat{}_{i+1}$. Depending on how closely the

actual cycle time follows the maximum cycle time, this step in the process can be repeated until $E_i_shifted\hat{}_{i+1}$ is close or equal to zero. The maximum cycle time will then follow that specified by the preliminary criterion when $E_{i+1}_shifted\hat{}_{i+1}$ is equal to zero and the process is repeated.

By symmetry, this process applies in the same way to control $E_i_shifted\hat{}_i$. In fact, this process can be applied equally to every element in the system since equations 6.29, 6.32 apply to $element_i$ and 6.31, 6.33 apply to $element_{i+1}$ where $element_i$ and $element_{i+1}$ (i=1..n-1) are any pair of neighbouring elements in the system. The maximum cycle time expressions according to the criterion have a common form, which is:

$$E_i_max_ct_j = E_i_max_p_ct_j + E_i_\phi_j \qquad 7.9$$

where $E_i_\phi_j$ is a positive quantity allowed in addition to the maximum cycle time specified by the preliminary criterion. The amount of $E_i_\phi_j$ imposes the maximum possible increase in $iE_i_late_j$. The following is the maximum cycle time expression under the criterion for reducing $E_i_shifted\hat{}_i$:

$$\begin{aligned} E_i_max_o_ct_j = {} & E_i_max_{po}_ct_j + E_i_shifted_{j-1}\hat{}_i \\ & + iE_{i+1}_total_input_delay_{k-1} - iE_i_total_output_delay_{j-1} \end{aligned} \qquad 7.10$$

Since

$$E_i_routput_delay_j = E_i_ct_j - E_i_max_{po}_ct_j \qquad 7.11$$

and if $E_i_ct_j$ equals $E_i_max_o_ct_j$, then according to equations 7.10, 7.11 and the definition of $iE_i_late_{jk}$, the following is true:

$$iE_{i}_late_{jk} = E_{i}_shifted_{j-1}\hat{}_{i} \qquad 7.12$$

which is the desired result for setting $E_{i+1}_shifted_{k}\hat{}_{i}$ to zero in equation 7.5. Likewise, to reduce $E_{i+1}_shifted\hat{}_{i+1}$, the maximum cycle time expression is:

$$E_{i+1}_max_{i}_ct_{k} = E_{i+1}_max_{pi}_ct_{k} + E_{i+1}_shifted_{k-1}\hat{}_{i+1} + iE_{i}_total_output_delay_{j-1} - iE_{i+1}_total_input_delay_{k-1} \qquad 7.13$$

Since

$$E_{i+1}_input_delay_{k} = E_{i+1}_ct_{k} - E_{i+1}_max_{pi}_ct_{k} \qquad 7.14$$

and if $E_{i+1}_ct_{k}$ equals $E_{i+1}_max_{i}_ct_{k}$, then according to equations 7.8, 7.9 and the definition of $iE_{i+1}_late_{k}$, the following is true:

$$iE_{i+1}_late_{k} = E_{i+1}_shifted_{k-1}\hat{}_{i+1} \qquad 7.15$$

which is the desired result for setting $E_{i}_shifted_{j}\hat{}_{i+1}$ to zero in equation 7.7.

Rewriting equation 7.13 for element$_{i}$, the maximum cycle time under the criterion therefore is:

$$E_{i}_ct_{j} \leq E_{i}_max_ct_{j} = \min(E_{i}_max_{i}_ct_{j}, E_{i}_max_{o}_ct_{j}) \qquad 7.16$$

such that the criterion can be satisfied by element$_{i}$ between element$_{i-1}$ and element$_{i}$, and between element$_{i}$ and element$_{i+1}$.

7.2.1.2.2 Analysing the Criterion

There is one concern about the property of iE_late (of any element) which may affect the results when the criterion is applied. Once iE_late is increased by a positive ideal world delay, it will continue to have that value for an unknown amount of time,

according to its definition and since the value of an ideal world delay will be included in the total ideal world delay term.

Suppose iE_{i+1}_late of equation 7.7 has been increased to a positive value which is close to E_{i+1}_shifted$\hat{}_{i+1}$, as described by the criterion. If the situation described by equation 7.7 occurs, then E_i_shifted$\hat{}_{i+1}$ is reduced as desired. Suppose E_i_shifted$\hat{}_{i+1}$ is now close to zero and if the situation described by equation 7.8 occurs, then E_{i+1}_shifted$\hat{}_{i+1}$ will be close to iE_{i+1}_β. Suppose iE_{i+1}_β is less than iE_{i+1}_late, and if the situation described by equation 7.7 occurs again, then E_i_shifted$\hat{}_{i+1}$ would become negative. This is the problem the application of the criterion tried to solve in the first place.

The problem with the application of the criterion is that in summary, a negative value may result in E_i_shifted$\hat{}_{i+1}$, if during the time iE_{i+1}_late for element$_i$ has an increased value, the situations described by equations 7.7 and 7.8 occur alternatively. The problem can happen equally to element$_i$, which is that in summary, a negative value may result in E_{i+1}_shifted$\hat{}_i$, if during the time iE_i_late for element$_{i+1}$ has an increased value, the situations described by equations 7.5 and 7.6 happen alternatively. These are summaries of the problems associated with the application of the criterion to element$_i$ and element$_{i+1}$.

To study the current problem with the process, the unknown duration of iE_late while it stays at an increased value is examined. In considering a scenario involving element$_i$ and element$_{i+1}$, the definition of iE_i_late for element$_{i+1}$ and iE_{i+1}_late for element$_i$ takes a general form as follows:

$$iE_s_late = iE_s_rio_delay_j + iE_s_total_io_delay_{j-1} - iE_t_total_io^{-}_delay_{k-1} \quad 7.17$$

where s = i or i+1 and io = input or output. If s = i then t = i+1 and io = output, io$^-$ = input. If s = i+1 then t = i and io = input, io$^-$ = output.

Initially, iE_s_total_io_delay and iE_t_total_io$^-$_delay are zero. In order to have a positive value for iE_s_late, iE_s_rio_delay is increased to a positive value. After the current cycle, the value of iE_s_rio_delay will be included in iE_s_total_io_delay. Therefore, this increased value will remain in effect until it is cancelled by an increased iE_t_total_io$^-$_delay. However, when this cancellation occurs is an unknown.

Fortunately, according to the above general expression and the fact that E_i_output_wait as described by equation 6.29 and E_{i+1}_input_wait as described by equation 6.31 cannot happen during the same cycle, an increase in iE_i_late of equation 6.31 will result in a decrease in iE_{i+1}_late of equation 6.29, and vice versa. Moreover, between element$_i$ and element$_{i+1}$, the chance of having iE_i_shifted$^\wedge_{i+1}$ be negative should equal the chance of having iE_{i+1}_shifted$^\wedge_i$ be negative, since the maximum cycle time criterion is applied equally to both elements. Therefore, the problem of having iE_i_shifted$^\wedge_{i+1}$ or iE_{i+1}_shifted$^\wedge_i$ be negative due to an increased iE_{i+1}_late or iE_i_late respectively, should be corrected by itself if given a long enough period of time.

For example, between element$_i$ and element$_{i+1}$, if a real world output wait (E_i_output_wait) as described by equation 6.29 has happened and iE_{i+1}_late is positive so that E_i_shifted$^\wedge_{i+1}$ is reduced to be close to zero, then E_i_shifted$^\wedge_i$ will very likely be positive. Following the argument of the last paragraph and if it is given that iE_{i+1}_late of equation 6.29 is positive then iE_i_late of equation 6.31 must be negative. iE_i_late will

then very likely be increased since the criterion tries to equally achieve the situations described by equations 7.12 and 7.15 and therefore, decreasing iE_{i+1}_late. The decrease of iE_{i+1}_late in this situation is desirable since it prevents the problem described in this section from happening. The remaining question is how soon this self-correction of the problem takes place.

Firstly, the described problem will exist if the situations described by equations 6.29 and 6.31 happen alternatively. The situations described by equations 6.29 and 6.31 are the real world output wait and the real world input wait, respectively, between two elements. With the existence of inter-stage storages, the real world output wait and the real world input wait happening alternatively between two elements must be separated by a certain amount of time.

Secondly, continuing the example of the last paragraph, after a real world output wait as described by equation 6.29 has occurred, the real world storage between $stage_i$ and $stage_{i+1}$ is full. This requires $stage_i$ to be slower than $stage_{i+1}$ for a certain amount of time in order to empty the storage and have a real world input wait as described by equation 6.31. It is therefore very likely that $stage_i$ will have positive ideal world delays which thereby increase iE_i_late and decrease iE_{i+1}_late before a real world input wait as described by equation 6.31 can happen and thus avoiding the described problem. Moreover, the occurrence of the situations described by equations 6.32 and 6.33 also decreases the likelihood of the described problem from happening.

The above example can be applied equally to the situation where E_{i+1}_shifted^$_i$ is reduced to be close to zero after a real world input wait as described by equation 6.31 has occurred. Based on the above arguments, it is reasonable to assume that the problem

associated with the application of the criterion, as described in this section, is a self-correcting one. The validity of this assumption will be demonstrated when emulation examples are presented in later sections.

The criterion of this section is ready to be applied to every element in the system, although its derivation is concentrated only on element$_i$ and element$_{i+1}$. This can be verified by expanding equations 6.29 to 6.33 for an additional number of elements, such as element$_{i-x}$ to element$_{i+y}$ ($1 \leq i\text{-}x \leq i+y \leq n$) consecutively. By examining the expanded equations, it can be seen that the criterion minimizes each term individually, such as those described by equation 7.7 for $E_i_shifted\hat{}_{i+1}$, while the assignment of terms such as those described by equation 7.8 for $E_{i+1}_shifted\hat{}_{i+1}$ can have two instances, for an element which is surrounded by two other elements.

7.2.1.2.3 Satisfying the Lower Bound of Maximum Cycle Time Constraint

Another issue concerning the criterion is whether the lower bound of the maximum cycle time is as is required by the constraint. This requires the lower bound of the maximum cycle time specified by the criterion to be the same as that specified by the preliminary criterion. However, if $E_i_\phi_j$ of equation 7.9 is negative, then $E_i_max_ct_j$ is less than $E_i_max_p_ct_j$. The constraint of the lower bound of the maximum cycle time can then be violated. $E_i_\phi_j$ is negative if the following is true for element$_i$, according to equation 7.10:

$$E_i_shifted_{j-1}\hat{}_i + iE_{i+1}_total_input_delay_{k-1} < iE_i_total_output_delay_{j-1} \quad 7.18$$

This can happen if ($E_i_shifted_{j-1}\hat{}_i < 0$) and/or ($iE_{i+1}_total_input_delay_{k-1} <$

$iE_i_total_output_delay_{j-1}$).

If ($E_i_shifted_{j-1}\hat{}_i < 0$), meaning the problem described in the last section has occurred, the solution should follow that which has been described. The minimization process of imposing an $E_i_max_o_ct_j$ to satisfy equation 7.10 is then not necessary. However, $iE_i_late_{jk}$ should not be increased since this would worsen the problem described in the last section and slows down the self-correcting ability of the problem.

If ($iE_{i+1}_total_input_delay_{k-1} < iE_i_total_output_delay_{j-1}$), then this situation cannot be reversed by imposing an $E_i_max_o_ct_j$. However, $E_i_max_o_ct_j$ should not be greater than $E_i_max_{po}_ct_j$ in order to avoid an increase in the inequality.

Based on these arguments, $E_i_max_o_ct_j$ is set to equal $E_i_max_{po}_ct_j$ if $E_i_\phi_j$ of equation 7.9 is negative; otherwise, the expressions for the maximum cycle time criterion are used. This argument applies equally to $E_{i+1}_max_i_ct_k$. Therefore, the lower bound of the maximum cycle time as required by the constraint is satisfied by the criterion.

7.2.1.2.4 The Output Stage Under the Criterion

The criterion as stated in equation 7.16 keeps $E_i_shifted\hat{}_c$ ($i=1..n\ c=1..n$), close to zero. The objective of the 'maximum cycle time algorithm' as stated in section 7.2.1 is to minimize production loss by minimizing ($iE_n_total_output_delay_j$ - $E_n_shifted_j$). Therefore, this section extends the existing criterion for the output stage, so that a positive $E_n_shifted_j\hat{}_n$ can result to offset the value of $iE_n_total_output_delay_j$.

The value of $iE_n_total_output_delay_j$ is inevitably positive under the current criterion, since the criterion allows maximum cycle times to be greater than those specified by the preliminary criterion. $iE_n_total_output_delay_j$ is an ideal world

measurement of element$_n$, which is independent of any of the other elements in the system. Therefore, it is only fair to the other elements if the value of $iE_n_total_output_delay_j$ is offset by the effect of element$_n$ alone. The effect of element$_n$ on the system output is $E_n_shifted_j\hat{}_n$. Thus by making $E_n_shifted_j\hat{}_n$ equal to $iE_n_total_output_delay_j$, the maximum cycle time algorithm is complete.

If a new value of $E_n_\phi_j$ for element$_n$ is used which is less than the one specified by the criterion then a positive $E_n_shifted_j\hat{}_n$ can result. The resulting $E_n_shifted_j\hat{}_n$ should be equal to the difference between this new value of $E_n_\phi_j$ and that specified by the criterion after enough time has passed. Therefore, in order to have $E_n_shifted_j\hat{}_n$ equal $iE_n_total_output_delay_j$, $E_n_\phi_j$ is replaced by $(E_n_\phi_j - iE_n_total_output_delay_j)$ in the maximum cycle criterion for element$_n$. The extension of the criterion is the new definition of the maximum cycle time for stage$_n$, and is stated as follows:

$$E_n_max_i_ct_k = E_n_max_{pi}_ct_k + E_n_shifted_{k-1}\hat{}_n + iE_{n-1}_total_output_delay_{j-1} - iE_n_total_input_delay_{k-1} - iE_n_total_output_delay_k \quad 7.19$$

$E_n_max_o_ct$ is not applicable since a downstream stage of stage$_n$ does not exist, and $E_n_output_wait$ would not occur under the current assumption of the system.

The algorithm applies the criterion specified by equation 7.16 to the first of the n-1 elements and applies equation 7.19 to the system output element. To intuitively explain how all of the elements are treated equally under the algorithm, it is first assumed that equation 7.16 is applied to all of the elements in the system. The overall effect of equation 7.16 is to equalize E_i_late for element$_{i+1}$ and E_{i+1}_late for element$_i$, while allowing both to increase in a controlled way. In other words,

iE_i_total_output_delay and iE_{i+1}_total_input_delay of any two neighbouring elements are being equalized. If all of the elements in the system satisfy equation 7.16, they would maintain a pace which is relative to the others, but the pace is not specifically set. However, this can lead to an unacceptable production loss if all of the elements are working much slower than they are supposed to while maintaining a relative pace. Therefore, one of the elements in the system must set the pace. The specified pace will then be followed by all of the other elements when equation 7.16 is applied to all of the other elements.

The output element is chosen to set the pace since minimizing system production loss is the goal. Therefore, equation 7.19 is derived for $element_n$. The extra limitation on the maximum cycle times given by equation 7.19 to $element_n$ is then carried throughout the system to each element by equation 7.16. The extra limitation is first experienced by $element_n$, then by $element_{n-1}$ through iE_n_total_input_delay, and so on. Intuitively, it can be understood that although different criterion are applied to $element_n$ and the other elements, the same limitation is experienced by all of the elements in the system. Therefore, the extended criterion should not violate the constraint that all of the elements in the system are treated equally.

7.2.1.2.5 Effect of Propagated Output Wait on the Criterion

This section discusses the effect the propagated output wait has on the application of the criterion. So far, the effect of propagated output wait has been ignored in the derivation of the criterion. A propagated output wait caused by $stage_L$ and effecting E_i_$shifted_j$^ has been described by equation 6.34. The i^{th} and L^{th} rows of this equation

are depicted here:

$$E_i_shifted_j\hat{}_i = iE_i_\alpha_j + E_L_shifted_{p-1}\hat{}_i \qquad 7.20$$

$$E_i_shfited_j\hat{}_L = -iE_L_late_{pm} + E_L_shifted_{p-1}\hat{}_L \qquad 7.21$$

These two expressions are similar to those considered when the criterion was derived. The results for both cases should be non-negative and follow the minimization process of the criterion. Therefore, the effects of other non-zero rows of equation 6.34 are considered here. They have been summarized as follows:

$$-(iE_\alpha_routput_wait_\mu + iE_\alpha_late_{\mu\lambda}) \qquad \alpha = i+1..L-1 \qquad 7.22$$

where $element_\alpha$ causes $E_{\alpha-1}_output_wait_\lambda$ by $iE_\alpha_late_{\mu\lambda}$. As stated earlier, one of the consequences of applying the criterion is the equalization of the total ideal world delays between two neighbouring elements. Assuming $iE_{\alpha-1}_total_output_delay_{\lambda-1}$ equals $iE_\alpha_total_input_delay_{\mu-1}$, then, according to the definition of $iE_\alpha_late_{\mu\lambda}$, equation 7.22 can be reduced to:

$$-(iE_\alpha_routput_wait_\mu + iE_\alpha_rinput_delay_\mu) \quad \alpha = i+1..L-1 \qquad 7.23$$

According to the definitions of the two terms in expression 7.23 which are contained in chapter four, the following is true:

$$iE_\alpha_routput_wait_\mu + iE_\alpha_rinput_delay_\mu = mct[(iE_\alpha_ob_{\mu-1} + iE_\alpha_ib_{\mu-1}) - (ob_limit - ib_limit - i)] + \theta_{\mu-1} \qquad 7.24$$

Also, from chapter four, recall that:

$$\begin{aligned} ob &\leq ob_limit + 1 \\ ib &\leq ib_limit \end{aligned} \qquad 7.25$$

Therefore, the following is true:

$$iE_{\alpha}_routput_wait_{\mu} + iE_{\alpha}_rinput_delay_{\mu} = mct \times \gamma + \theta_{\mu-1} \qquad 7.26$$

where γ is non-positive.

The value of $\theta_{\mu-1}$ is non-zero only if $iE_{\alpha}_routput_delay_{\mu-1}$ is greater than zero. Its maximum value is the output transfer time inside $element_{\alpha-1}$, which is a part of the transfer time between $stage_{\alpha-1}$ and $stage_{\alpha}$. The expression in 7.23 has a maximum negative value of $\theta_{\mu-1}$ and a maximum positive value of (γ x mct).

This result shows that as long as the criterion is minimizing the difference between $iE_{\alpha-1}_total_output_delay_{\lambda-1}$ and $iE_{\alpha}_total_input_delay_{\mu-1}$, the negative effect of a propagated output wait on an element is very limited. Therefore, the effect of a propagated output wait has been ignored in the derivation of the criterion. However, if the negative effect of a propagated output wait on the E_i_shifted of an element becomes noticeable due to the unbalancing of $iE_{\alpha-1}_total_output_delay_{\lambda-1}$ and $iE_{\alpha}_total_input_delay_{\mu-1}$, the performance of the maximum cycle time algorithm deteriorates such that some of the rows of E_i_shifted^ become negative. For the current implementation of the algorithm, no extra step has been taken to prevent such a negative effect from happening. This negative effect, if it ever takes place, is assumed to be corrected by the self-correcting property of the criterion. Theoretically, if given enough time, the negative effect tends to become zero or positive again, following the argument used in the derivation of the criterion.

7.2.1.3 Summary

The maximum cycle time algorithm has been summarized in a block diagram in figure 7.10 at the end of this chapter. The maximum cycle time algorithm is executed at the beginning of each cycle for each stage for two reasons. Firstly, all of the information required by the maximum cycle time calculation for a cycle of a stage is already available at the beginning of that cycle and secondly, in a practical situation, the maximum cycle time limit for a cycle will be much more valuable if it is available at the beginning rather than at the end of that cycle. Moreover, the maximum cycle time for a cycle of a stage calculated at the beginning of the cycle will have the minimum value among all of the values calculated at different times during the same cycle. This is because (iE_{i+1}_total_input_delay) of equation 7.10, (iE_i_total_output_delay) of equation 7.13 and (iE_{n-1}_total_output_delay) of equation 7.19 can only increase while all of the other terms in these three equations remain the same during the same cycle. Therefore, in a practical situation, the maximum cycle time for a cycle of a stage can be recalculated if any of the above three variables have changed during a cycle and the updated maximum cycle time value will not be less than the one calculated previously during the same cycle for the stage.

7.2.2 Emulation Examples

The following examples show the effect of applying the maximum cycle time algorithm to emulated systems. The results of these examples try to show:

1. the difference between the production times before and after the application of the algorithm for an amount of production volume.

2. the difference between the efficiencies before and after the application of the algorithm.

3. the validity of the assumptions made when deriving the criterion.

In order to illustrate the algorithm, it is assumed that the maximum cycle time calculated by the algorithm is observed by all of the stages, although this is not completely realistic in a real production environment.

7.2.2.1 A Three-Stage System Example

The first example shows a system with three stages. The maximum cycle time algorithm is implemented by a brute-force method using a batch mode emulator. The derivation of the criterion has assumed that the values of all of the required variables, like the total ideal world delays, are available and up to date. However, in a real situation, these variables are changing as long as there is a job being finished by a stage. In the implementation of this algorithm in a real environment, from the point of view of a particular stage, the maximum cycle time calculated by the algorithm will be increased when a new job is finished by any of its neighbouring stages. This can be verified by examining equations 6.29 to 6.33, and the cumulative property of the total ideal world delays. Therefore, from the point of view of a stage, as long as the cycle time used to produce the current job is no greater than the most up to date maximum cycle time, the criterion is satisfied.

To ensure the maximum cycle time criterion is satisfied by a stage in emulation, its cycle time is checked with the maximum cycle time calculated based on the information received just before end of the cycle. If the cycle time is greater than the

calculated maximum cycle time, it is replaced by the calculated maximum cycle time. However, this off-line change of cycle times in a batch emulator requires re-running the emulation at some point in time before the end of that particular cycle. The re-run is required because the change of a cycle time for one stage changes many events in the system before and after the change of cycle time takes place. Therefore, the starting point of the re-run should be earlier than any of the events which are changed because of the change of cycle time. For simplicity's sake, a brute-force method is used in this example so that the emulation is re-run from the beginning whenever a cycle time changes.

In this example of a three stage system, the size of each inter-stage storage is one unit. Initially, all of the inter-stage storages are empty, and there is a job inside each stage. The cycle times of each stage follow a Gamma distribution which is shown in figure 7.1, which the author believes is a reasonable assumption about the performance of human workers. The mean processing cycle time is arbitrarily set to 60 time units.

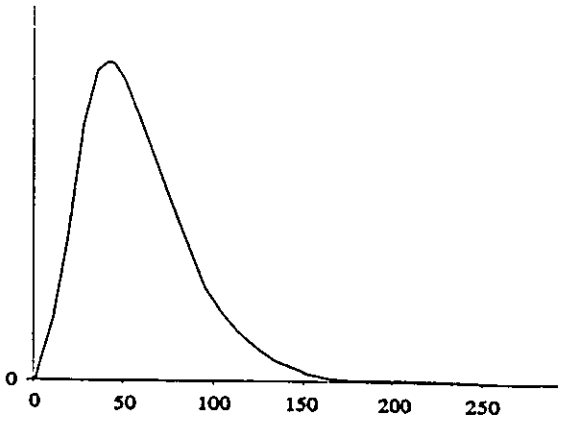


Figure 7.1 Cycle time probability density function

The run time for this example is determined by the number of jobs to be produced, which has been chosen to be 20,000. Table 7.1 shows the mean cycle time, standard deviation, minimum and maximum cycle times used by the three stages.

	Mean	Std. dev.	Minimum	Maximum
$Stage_1$	59.9	29.9	1.5	294.3
$Stage_2$	60.0	29.8	1.5	250.8
$Stage_3$	60.0	30.0	2.1	316.7
Average	60.0	29.9	1.7	287.3

Table 7.1 Cycle time statistics before applying the MCT algorithm

Figure 7.2 shows $E_3_shifted\hat{}_i$ (i=1..3), before the maximum cycle time (MCT) algorithm is applied. The ideal world total output and input delays between two

neighbouring elements are not shown in this case since they are predictable according to this diagram, as was the case for the examples contained in chapter six.

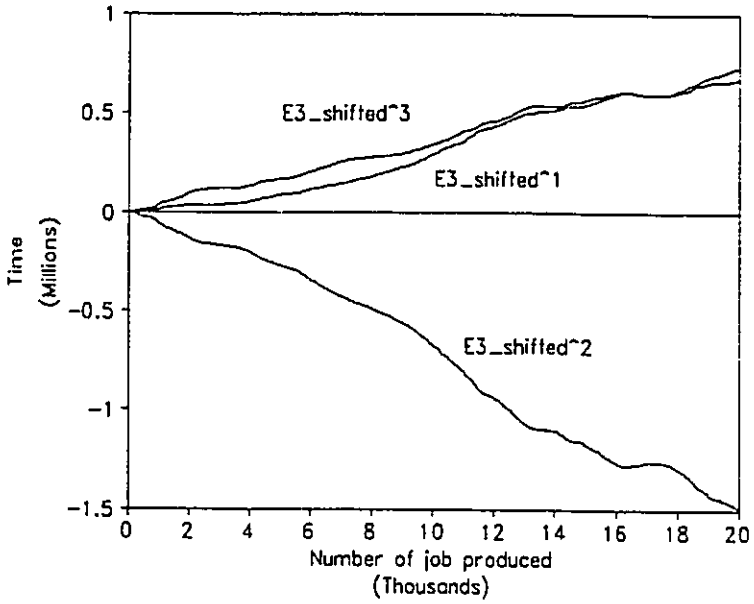


Figure 7.2 E_3_shifted^$_i$ i=1..3, vs. number of jobs before applying the MCT algorithm

The production time for 20,000 jobs in this case is 1422798.8 time units. Using the efficiency definition contained in chapter five, the efficiency of the system in this case is 84.3%.

Figure 7.3 shows E_3_shifted^$_i$ (i=1..3), after the maximum cycle time algorithm has been applied. Notice that E_3_shifted^$_1$ and E_3_shifted^$_2$ look like they are close to zero in this scale of the diagram, which is the desired result when the maximum cycle time algorithm is applied. E_3_shifted^$_3$ on the other hand is increasing throughout the production period. The positive result of E_3_shifted^$_3$ should be close to

iE_n_total_output_delay, so that they can cancel each other, as described in equation 7.19.

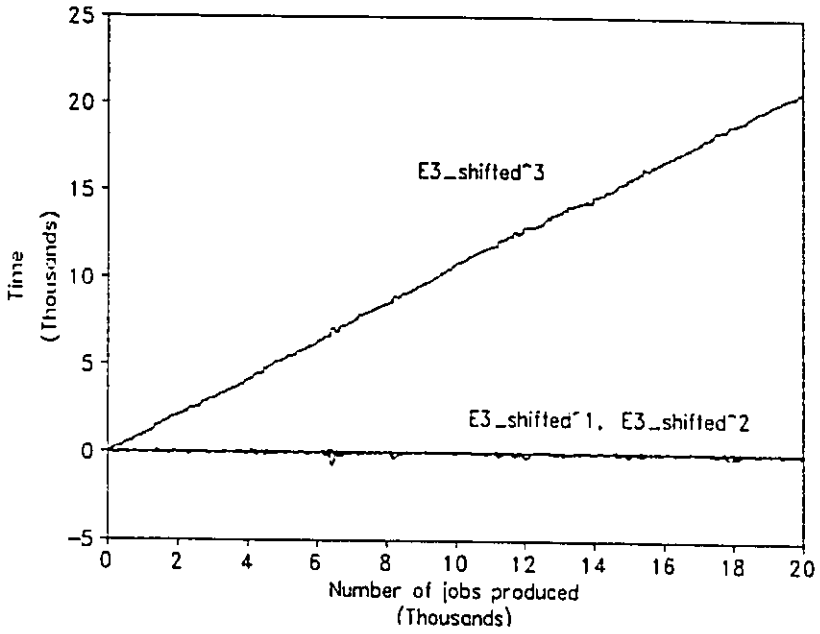


Figure 7.3 E_3_shifted^$_i$ i=1..3, vs. number of jobs after applying the MCT algorithm

Figure 7.4 shows iE_n_total_output_delay on the same scale. On this scale of the diagram, E_3_shifted^$_3$ and iE_n_total_output_delay seem to be equal as desired.

The production time for 20,000 jobs in this case is 1200044.8 time units. The mean cycle time for each element is significantly lower in this case, since a number of the cycle times were trimmed by the maximum cycle time algorithm. This also led to the lower standard deviations which are shown in table 7.2. The average mean cycle time shown in table 7.2 is used to calculate the efficiency using the definition contained in chapter five. The efficiency of the system in this case increased from 84.3% to 90.5% after the MCT algorithm was applied.

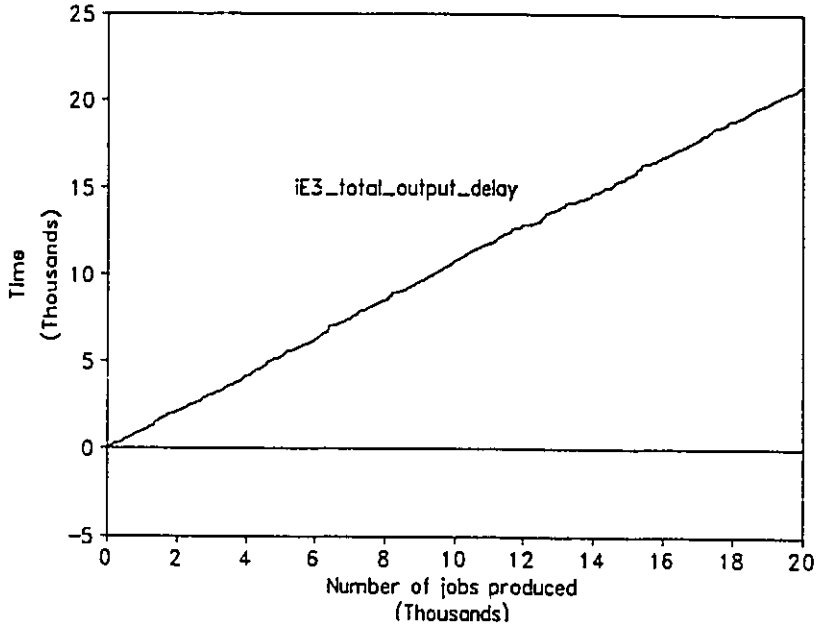


Figure 7.4 E_3_total_output_delay vs. number of jobs after applying the MCT algorithm

	Mean	Std. dev.	Minimum	Maximum
$Stage_1$	54.3	21.6	1.5	170.6
$Stage_2$	54.3	21.7	1.5	193.7
$Stage_3$	54.3	21.9	2.1	177.4
Average	54.3	21.7	1.7	180.6

Table 7.2 Cycle time statistics after applying the MCT algorithm

7.2.2.2 Discussion of The Three-Stage Example

The results of this example came very close to the objective such that the production time for producing 20,000 jobs after applying the algorithm is within 0.0037% of the expected time. The assumptions used to derive the algorithm seemed to be satisfactory in this case, as it is shown in figure 7.3 that $E_3_shifted\hat{}_1$ and $E_3_shifted\hat{}_2$ were very close to zero most of the time. Although in figure 7.3, $E_3_shifted\hat{}_1$ and $E_3_shifted\hat{}_2$ were sometimes less than zero, they corrected themselves in a relatively short amount of time. On the other hand, the failure to self-correct would result in a problem that would collapse the algorithm, so that one or more of $E_3_shifted\hat{}_i$ would become and stay negative.

The instantaneous relationships and the deterministic calculations used in the algorithm makes it possible for on-line processing. For instance, it is possible in a production system for one stage to exceed the suggested maximum cycle time without significantly modifying its mean cycle time which is available. Usually, slight instantaneous variations of the mean cycle time are ignored because they are common. This algorithm points out that if this stage does not lower its cycle time to below the maximum cycle time, the rate of production will decrease.

The results of the above example show that a lower mean cycle time and cycle time standard deviation resulted after the algorithm was applied and a higher efficiency was achieved. The calculation of efficiency according to the definition contained in chapter five takes the lower mean cycle times into account. The lower cycle time deviations, however, were not encountered. The increase in efficiency therefore was achieved at the cost of the cycle time variations.

The argument that the maximum cycle time algorithm treats every element equally seems to be supported by table 7.2. The mean cycle times and standard deviations shown in that table for each element were acceptably 'close' to each other. The minimum and maximum cycle times were affected by their original values. Finally, iE_i_total_output_delay and iE_{i+1}_total_input_delay for i=1,2 after the application of the MCT algorithm as shown in figure 7.5 and 7.6 were as expected as their rates of increase were close to each other's. In these two figures, since the initial values of iE_i_total_output_delay and iE_{i+1}_total_input_delay were both zero, they seemed to equal each other.

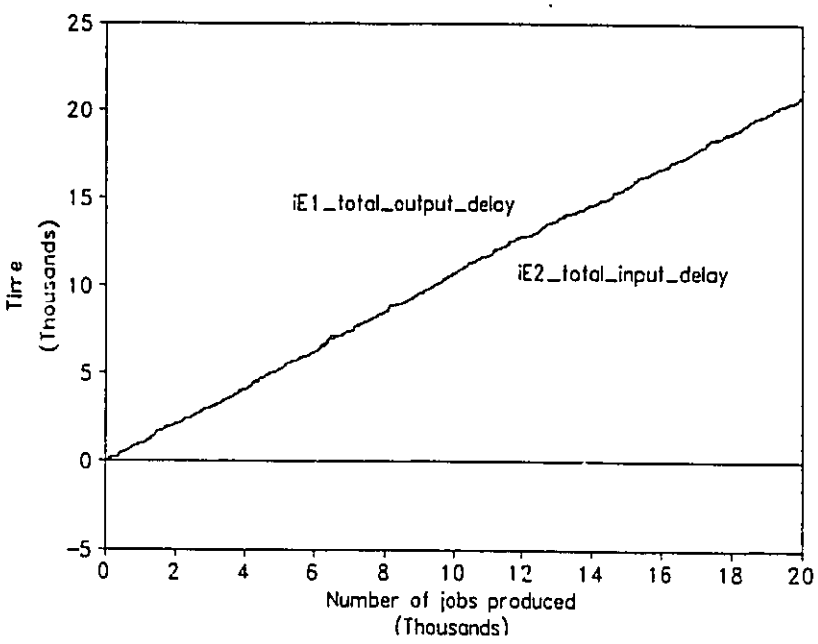


Figure 7.5 iE_1_total_output_delay and iE_2_total_input_delay after the application of the MCT algorithm

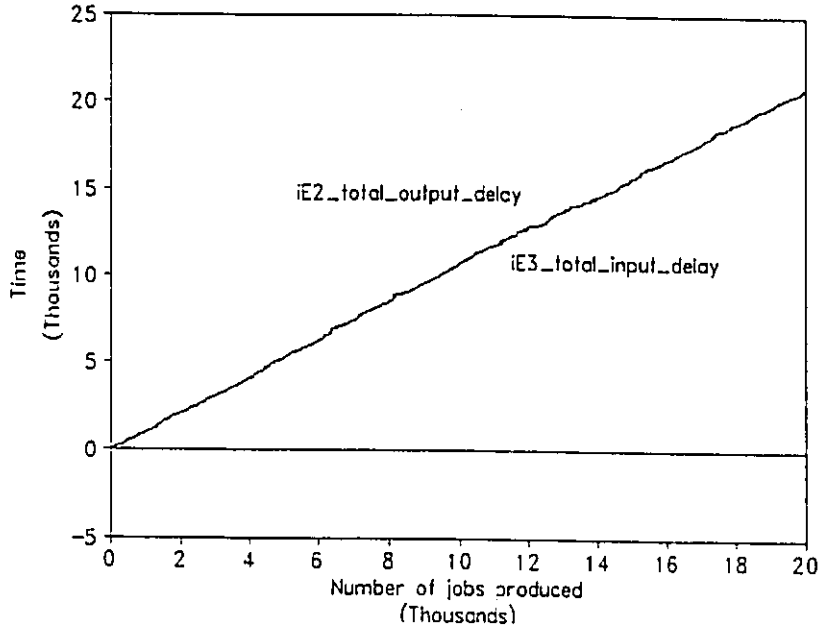


Figure 7.6 iE_2_total_output_delay and iE_3_total_input_delay after the application of the MCT algorithm

7.2.2.3 Ten-Stage System Examples

A ten-stage system is emulated in the following examples. The brute-force method of implementation using a batch emulator for the maximum cycle time algorithm is inappropriate in this case because the total run time for the emulation, including all of the re-runs in this case, would be an exponential increase over that of the previous example. Therefore, an interactive emulator is used so that the emulated system can be restored to an earlier point in time and from there a re-run can start. The reason for requiring re-runs has been discussed in the previous example and the point in time where a re-run starts is determined analytically in this case.

In the following examples, the system setup follows that of the previous example.

The size of each inter-stage storage is one unit, and initially, all of the inter-stage storages are empty, and there is a job inside each stage. The same cycle time distribution, as was shown in figure 7.1, is used. The number of jobs to be produced is 20,000.

In appendix A, Table A.1 shows the cycle time statistics for the ten stages, and figures A.1 to A.3 show $E_{10}_shifted\hat{}_i$ (i=1..10) before the maximum cycle time algorithm is applied. The production time for producing 20,000 jobs before applying the maximum cycle time algorithm is 1546685.1 time units. The efficiency of the system in this case is 77.6%.

In appendix B, figure B.1 to B.3 shows $E_{10}_shifted\hat{}_i$ (i=1..10) after the maximum cycle time algorithm is applied, as described in section 7.2.1. Table B.1 shows the cycle time statistics. Notice that after the maximum cycle time algorithm is applied in this case, $E_n_shifted\hat{}_i$ (i=1..n-1), are not as close to zero as in the case of the three stage system example. There are noticeable violations of the assumptions made in the algorithm derivation since there are noticeable times when $E_n_shifted\hat{}_i$ (i=1..n-1), are negative. Although they tend to correct themselves and move back to the positive region when given enough time, the amount of time required to do this though is much longer than that in the three-stage system case. Therefore, improving the maximum cycle time algorithm can be useful. Figure B.4 shows $iE_{10}_total_output_delay$ which closely matches $E_{10}_shifted\hat{}_{10}$ as was the case with the three-stage system example.

The production time for producing 20,000 jobs after the maximum cycle time algorithm is applied is 1210403.4 time units, which is within 1% of the expected time for producing this number of jobs. The mean cycle time for each element is again

significantly lower after the algorithm is applied. The system efficiency calculated using the lower mean cycle time after the algorithm has been applied in this case increased from 77.6% to 88.4%. As with the three-stage system case, this increase in efficiency is at the expense of a lower standard deviation in cycle times which is not accounted for in the efficiency calculation. The mean cycle times and standard deviations shown in table B.1 again show that the algorithm treats every element fairly equally. However, this example shows a slight degradation in the performance of the algorithm. A modification of the algorithm is therefore attempted.

7.2.3 The Minimization Factor

The results of the three-stage and the ten-stage examples agree with the hypothesis that the number of stages in a system is one of the factors which determine system efficiency. This is shown by comparing the results of the two examples obtained both before and after the maximum cycle time algorithm is applied. In the case of the maximum cycle time algorithm application, the number of stages in a system also affects the stability of the results. The resultant $E_n_shifted_i$ ($i=1..n-1$), after the algorithm is applied in both cases fluctuated above and below the zero line, although the ideal result should be equal to or above zero. However, the extent of the fluctuation is more intense with the results of the ten-stage case than those of the three-stage case. Therefore, a compensation method for this effect has been derived, so that the undesired results of the application of the algorithm to a larger system is reduced to be closer to that of a smaller system. In other words, the compensation method should reduce the inter-dependency between the extent to which, after applying the algorithm, the resulting $E_n_shifted_i$

(i=1..n-1), fluctuates below zero, and the size of the system.

The problem addressed in the section 'analysing the criterion' has been assumed to be a self-correcting one. However, from the results of the ten-stage example, the validity of the assumption becomes uncertain. The duration of time which some of the $E_{10}_shifted\hat{}_i$ stayed negative is a clear indication of the weakness of the assumption, although the resulting production time is about 1% of the expected time for producing the specified amount of job. Therefore, a slight variation will be introduced in this section to strengthen the validity of the assumption under this circumstance.

The 'Minimization Factor' (MF) reduces the rate at which the maximum cycle time algorithm decreases $E_i_shifted\hat{}_c$ (i=1..n, c=1..n), from positive to the desired value. The range of MF is usually from 0 to 1 inclusively. The effect of MF is to replace the effect of the original algorithm as described in equations 7.12 and 7.15 with the following, which applies to all i:

$$iE_i_late_j = MF \times E_i_shifted_{j-1}\hat{}_i \qquad 7.27$$

The rate at which $iE_i_late_j$ increases as allowed by the algorithm is reduced if MF is between 0 and less than 1 and therefore the rate at which $E_i_shifted_j\hat{}_i$ is decreased by the algorithm is reduced. The effect of MF delays the time when the problem described in the section 'analysing the criterion' takes place, thus leaving more time for the self-correcting action. The existence of a natural time delay for the problem to take place and the self-correcting action were discussed in section 7.2.1.2.1.

With the minimization factor in place, equations 7.10 and 7.13 of the criterion become:

$$E_{i+1}_max_i_ct_k = E_{i+1}_max_{pi}_ct_k + MF \times E_{i+1}_shifted_{k-1}\hat{}_{i+1} + iE_i_total_output_delay_{j-1} - iE_{i+1}_total_input_delay_{k-1} \quad 7.28$$

and

$$E_i_max_o_ct_j = E_i_max_{po}_ct_j + MF \times E_i_shifted_{j-1}\hat{}_i + iE_{i+1}_total_input_delay_{k-1} - iE_i_total_output_delay_{j-1} \quad 7.29$$

For the output stage, equation of 7.19 of the criterion becomes:

$$E_n_max_i_ct_k = E_n_max_{pi}_ct_k + MF \times (E_n_shifted_{k-1}\hat{}_n - iE_n_total_output_delay_k) + iE_{n-1}_total_output_delay_{j-1} - iE_n_total_input_delay_{k-1} \quad 7.30$$

so that $E_n_shifted\hat{}_n$ will equal $iE_n_total_output_delay$ in the long run.

When MF equals one, these equations are the same as those of the original algorithm. When MF equals zero, all of the total_output_delay and total_input_delay terms will be zero and the above equations are reduced to be equal to those of the preliminary criterion. If the maximum cycle time algorithm is applied according to the preliminary criterion, a positive value will very likely result in each $E_i_shifted\hat{}_c$ ($i=1..n$, $c=1..n$). On the other hand, some $E_n_shifted\hat{}_i$ ($i=1..n$), may be negative for a undesirably long period of time if the original maximum cycle time algorithm is applied. Therefore, the choice of MF should be a compromise between these two extremes according to the number of stages in the system.

The original ten-stage example shown in appendix A is applied with the maximum cycle time algorithm for different values of the minimization factor. The results for MF=0.75, 0.5 and 0.25 are presented in appendices C, D and E, respectively. The result of applying the maximum cycle time algorithm according to the preliminary

criterion is shown in appendix F for reference which is the same as having MF=0. The averages of some of the times concerned over the ten stages from each run are summarized below in table 7.3.

MF	Avg. Mean	Avg. S. D.	Avg. Min.	Avg. Max.	Err.(%)	Eff.(%)
∞	59.98	29.93	2.3	270.0	-28.9	77.56
1.00	53.47	20.37	2.3	186.2	-0.9	88.35
0.75	53.5	20.39	2.3	171.6	-0.3	88.91
0.50	53.55	20.43	2.3	143.3	0.0029	89.25
0.25	53.59	20.47	2.3	128.8	0.054	89.37
0.00	49.27	15.2	2.3	67.9	8.1	89.38

Table 7.3 Averages over the ten stages from different runs using different MFs

The results contained in the first row of the table where MF=∞, meaning the maximum cycle time limit is infinity, are equivalent to the results where the maximum cycle time algorithm has not been applied. Notice that the system efficiency shown in the last column of the table increases when MF decreases. The average maximum cycle time decreases with the decrease of MF. The error term in the second to last column is a measure of the percentage difference between the actual production time and the expected time for producing the specified number of jobs. Using MF=0.5 produces the lowest error among these cases. The efficiency of the system increased to 89.3% from 77.6% after the maximum cycle time algorithm is applied with MF=0.5. Again, this improvement is at the expense of a lower average standard deviation of the cycle times.

The original motivation for introducing the minimization factor has been justified,

as is shown in the plottings of E_{10}_shifted^i (i=1..10) for the different values of MF, which are shown in the appendices. As MF decreases, the possibility of having some of E_{10}_shifted^i (i=1..10) become negative decreases both in magnitude and duration, as desired. With the case of MF=0.25, the problem is almost non-existent. The magnitude of iE_{10}_total_output_delay matches the corresponding E_{10}_shifted^10 for all of the cases.

There is no further search for an optimal minimization factor with some pre-defined goal, since the search would be a trial and error process without any further analytical study of the matter. Furthermore, by observing the 'shifted' plottings contained in the appendices, it is conjectured that an optimal minimization factor should not be a constant. Instead, it should have a lower value for the stages in the middle and a higher value for the stages at the two ends of the system. The improvement achieved by this modified algorithm justifies the use of the minimization factor even though it adds a minor complexity to it. However, this is warranted by confidence in the algorithm in situations where there are a larger number of stages.

7.3 EXAMPLE II: A PRELIMINARY INVESTIGATION OF Storage ALLOCATION

Based on the current results obtained up to this point, it is interesting to see what information can be obtained for other problems concerning the system. The optimal storage allocation problem is chosen as an example because it is a well studied one. However, this preliminary study of the problem is by no means a complete one. Many details have been left out which need to be completed.

The optimal storage allocation problem has been defined generally as finding the amount of storage space between the stages which yields the maximum system efficiency for a certain type of cycle time distribution while satisfying a set of pre-defined constraints. Some of these constraints are, for example, the total storage space available (Sheskin 74), the facility cost and the cost associated with expected on-hand stock (Kraemer, Love 70). In this preliminary study, only the total available storage space will be of concern.

7.3.1 Long Term Ideal World Measurement

The basic argument in this study is that if the long term ideal world measurement of an element can be obtained, with the storages inside the element as variables, then the results obtained up to this point can be applied to help solve the problem. In this study the long term ideal world measurement of an element is approximated based on emulation results. Figure 7.7 shows the total ideal world delays of an element which has five input and five output storage units. The same cycle time distribution as the one used in the maximum cycle time example is used. Unlike the maximum cycle time example, the cycle times in this study are not monitored and changed once they are assumed. The ideal world total output delay and total input delay of an element are very close or equal to each other when the input and output storage spaces are equal. The diagram shown in figure 7.7 is obtained by using ten different sets of cycle times which have the same distribution. The initial content of both the output and input storages are set to be half full. The length of each run is the time it takes to produce 20,000 jobs.

Assuming the total ideal world delays can be approximated by a straight line, the

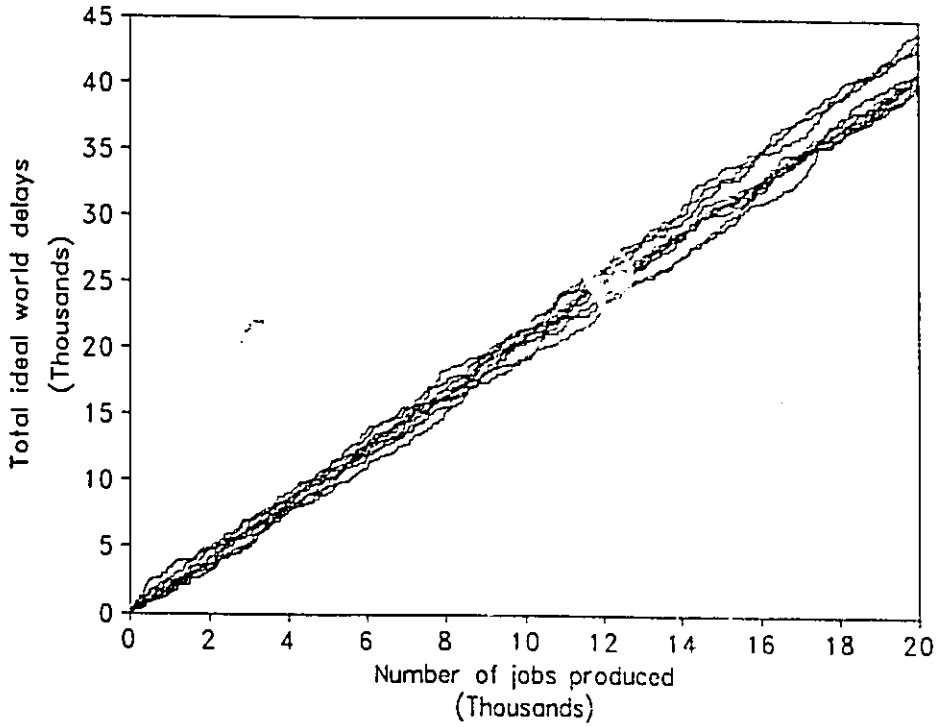


Figure 7.7 Total ideal world delays vs. number of jobs

total ideal world delays in figure 7.8 are then plotted for an element with different storage sizes. The ten cases in the diagram are for an element which may have one to ten storage units which are equal in input and output storage space. Case one is of an element which has one storage unit, case two of an element which has two storage units, and so on. In all of the cases, the total ideal world output delay is equal or very close to the total ideal world input delay. The straight line is an approximation of both of the total ideal world delays.

Using the same straight line approximation, the total ideal world delays in figure 7.9 are plotted for an element with different cycle time variations. The element used in the plotting has five output and five input storage units. The nine cases in the diagram are plotted using nine sets of cycle times which have the same mean cycle time but

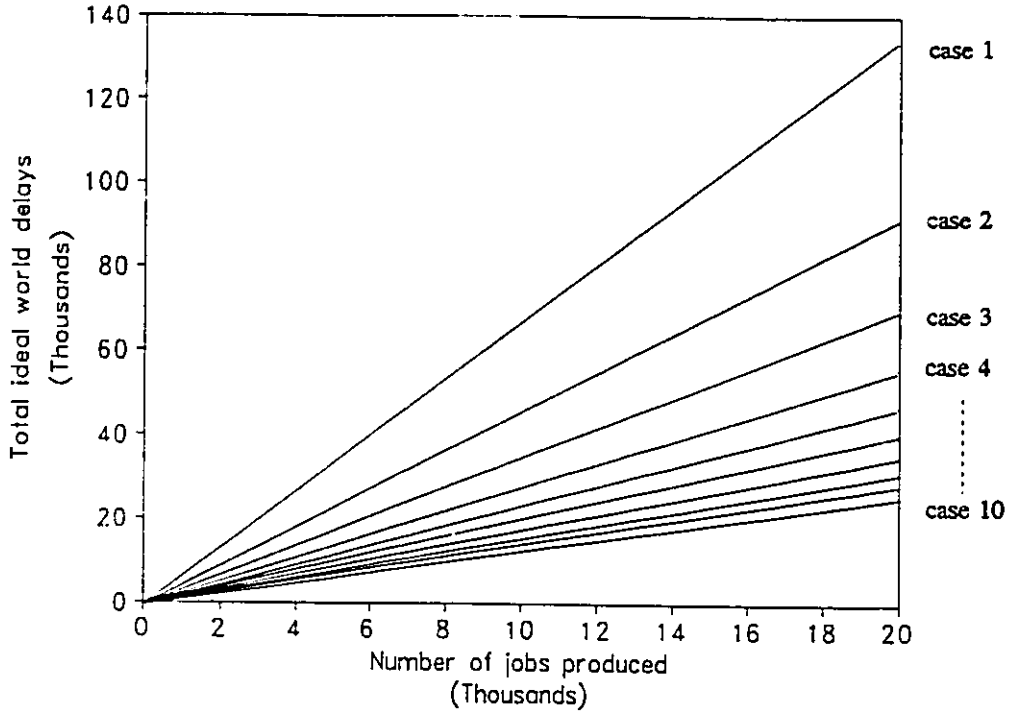


Figure 7.8 Approximate total ideal world delays for ten storage sizes

different standard deviations which are decreasing from case 1 to case 9 as shown in table G.1 of appendix G.

These two figures agree with the intuition that, for figure 7.8, the slope of the total ideal world delays decreases with the increase of the storage size and for figure 7.9, the slope of the total ideal world delays increases with the increase of the cycle time deviation. Assuming these relationships are always correct, the argument for storage allocation can be presented.

7.3.2 The Storage Allocation Argument

The current result has shown that there are three types of real world waits: the real world output wait, real world input wait, which are both between two neighbouring

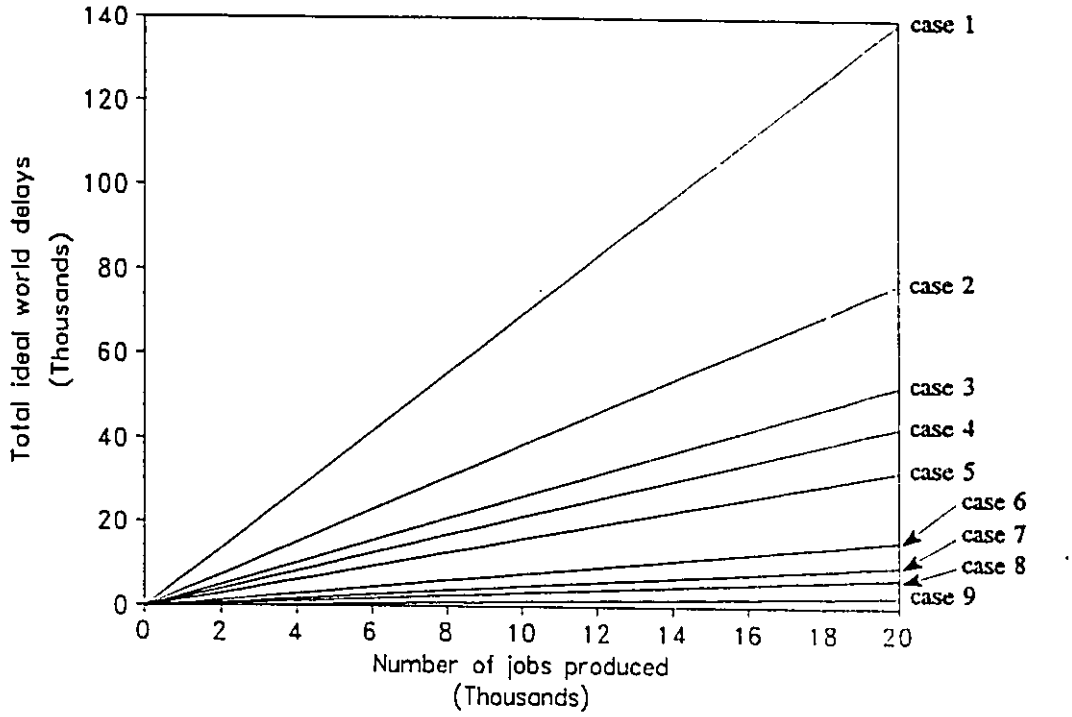


Figure 7.9 Approximate total ideal world delays for different cycle time deviations

stages, and the real world propagated output wait, which involves a series of consecutive stages. Any of these real world waits can contribute to production loss, which is shown by a negative value in $E_n_shifted_q\hat{}_i$ (i=1..n-1), and by $E_n_shifted_q\hat{}_i$ being less than $iE_n_total_output_delay_q$. The results also showed that in all three cases, a negative value in $E_n_shifted_q\hat{}_i$ (i=1..n) can be avoided by matching $iE_i_total_output_delay_j$ and $iE_{i+1}_total_input_delay_k$ between all of the two neighbouring elements, which forms the basic argument for the storage allocation problem. Therefore, the objective of this preliminary study of the storage allocation problem is to match $iE_i_total_output_delay_j$ and $iE_{i+1}_total_input_delay_k$, for i=1..n-1 in the system.

If the matching of the ideal world delays is achieved by choosing the right amount of inter-stage storages, the efficiency of the system should be close to optimal, using the

efficiency calculation contained in chapter five. This preliminary study suggests that the method to be used is as stated in the following paragraph.

Assign an equal amount of storage space to both the output and input sides of each element. The amount of storage space an element is assigned depends on the cycle time deviation of the stage inside the element so that the long term total ideal world delay plot of the element should match that of its neighbouring elements. The storage space between $stage_i$ and $stage_{i+1}$ in the real world is then the sum of the (iE_i_ob_limit + iE_{i+1}_ib_limit) of $element_i$ and $element_{i+1}$, according to the element definition. If the storage space of an element is a real number, then the sum should be rounded off to the nearest integer.

According to the assumption which is made based on the observations of diagrams 7.8 and 7.9, this approach agrees with the observation that "larger storage capacities should be allocated around machines of lower reliability" (Soyster et al. 79).

7.3.3 The Output Stage Under Storage Allocation

The matching of E_n_shifted$_q\hat{}_n$ with iE_n_total_output_delay$_q$ is considered in this section. The reason for the matching is explained in the maximum cycle time algorithm derivation. Using the straight line approximation, figures G.1, G.2 and G.3, which are shown in appendix G, are plotted for the total ideal world waits in the same way figures 7.7, 7.8 and 7.9 for the total ideal world delays were plotted. All of the settings such as the storage sizes and the cycle time deviations used for the total ideal world waits figures are the same as for the total ideal world delays figures. In figures G.1, G.2 and G.3, iE_i_total_output_wait is equal or very close to iE_i_total_input_wait when

iE_i_ob_limit equals iE_i_ib_limit. Thus the straight approximation is used for both of the total ideal world waits.

In the long run, the total ideal world waits behave very similarly to the total ideal world delays. By observing figures G.1 to G.3 and 7.7 and 7.9, it is assumed that the long term 'total ideal world wait' of an element equals its long term total ideal world delays. Assuming this property holds true in all of the cases, the following discussion can proceed.

If at least one of the ideal world waits for element$_i$ in cycle j is non-zero, such as (iE_i_output_wait$_j$ > 0) and/or (iE_i_input_wait$_j$ > 0), and the real world waits of stage$_i$ in cycle j are all zero, then E_i_shifted$_j$^$_i$ is increased according to equation 6.32 by:

$$\max(\ iE_i_output_wait_j,\ iE_i_input_wait_j\ ) \qquad 7.31$$

On the other hand, if at least one of the real world waits for element$_i$ in cycle j is non-zero, such as (E_i_output_wait$_j$ > 0) and/or (E_i_input_wait$_j$ > 0), then according to equations 6.29 and 6.30, E_i_shifted$_j$^$_i$ is increased by ($iE_i_\alpha_j$) or ($iE_i_\beta_j$), respectively, depending on which real world wait is dominating. Defining $\Sigma_\Delta E_i$_shifted$_j$^$_i$ to be the total increase of E_i_shifted$_j$^i since the beginning, then for stage$_n$, to which real world output wait would never occur, according to the definition of $iE_i_\beta_j$, the following is possible:

$$\Sigma_\Delta E_n_shifted_q\hat{}_n \geq iE_n_total_output_wait_q \qquad 7.32$$

If the negative effect on E_i_shifted^ is ignored by assuming iE_i_total_output_delay is matching iE_{i+1}_total_input_delay for i=1..n-1, then:

$$E_n_shifted_q\hat{}_n \geq iE_n_total_output_wait_q \qquad 7.33$$

If assuming that iE_i_total_output_delay equals iE_i_total_output_wait, then equation 7.33 is equivalent to the following:

$$E_n_shifted_q\hat{}_n \geq iE_n_total_output_delay_q \qquad 7.34$$

Equation 7.34 shows a possible scenario for the output element of a system since only the output element would never experience output waits under the current assumptions. The actual value of $\Sigma_\Delta E_n_shifted_q\hat{}_n$ depends on the order of occurrences of real world input waits of $element_n$, real world output waits of $element_{n-1}$, as described by equations 7.7 and 7.8, and its ideal world waits. On the other hand, iE_n_total_output_$delay_q$ of equation 7.34 can be reduced by increasing iE_n_ob_limit. However, these factors, which the values of $\Sigma_\Delta E_n_shifted_q\hat{}_n$ and $iE_n_total_output_delay_q$ depend upon, are not affected by the system storage allocation.

The discussion of this section shows that if the negative effect on E_i_shifted^ is ignored by assuming iE_i_total_output_delay is matching iE_{i+1}_total_input_delay for i=1..n-1, then the matching of $E_n_shifted_q\hat{}_n$ with $iE_n_total_output_delay_q$, or in a more general sense, making equation 7.34 to be true is possible. However, the likelihood that equation 7.34 is true does not depend on the system storage allocation process. Therefore, the matching of $E_n_shifted_q\hat{}_n$ with $iE_n_total_output_delay_q$ does not impose any new constraint on the method which has been suggested for storage allocation.

7.3.4 Summary of the Storage Allocation Method

When the total space for inter-stage storages is the only constraint imposed on

optimal storage allocation, the method introduced in this section can be used to obtain an approximate solution. The overall objective is to match iE_i_total_output_delay with iE_{i+1}_total_input_delay for i=1..n-1. The total storage space should be as close to the total available space as possible. The advantage of having the maximum possible total storage is that it reduces the slope of the total ideal world delays, as has been shown by figure 7.8. The consequence is that the slower the total ideal world delays increase, the lower the chance iE_i_total_output_delay will not follow iE_{i+1}_total_input_delay is, which violates the assumption supporting equation 7.33 and the overall objective.

The method introduced by this study has the advantage that it is already applicable to serial systems of any size. The reason supporting this method is based on the independent measurement of each element. Therefore, the same procedure can be carried out independent of the system size.

For future studies, the assumptions made in this section should be validated. Introduction of analytical methods for finding the long term ideal world measurements would be beneficial. Accompanying methods for fine tuning the approximate solution given in this section is recommended. These methods should first consider the negative effects which will arise if iE_i_total_output_delay does not follow iE_{i+1}_total_input_delay closely enough. Then, compensation techniques should be derived.

Finally, if the same analysis which was applied to the output stage in sub-section 7.3.3 is applied to the input stage, the result is:

$$\Sigma E_1_shifted_p\hat{}_1 \geq iE_1_total_input_wait_p \qquad 7.35$$

The above equation and equation 7.33 are possible because of the assumption that there

are infinite jobs in front of $stage_1$ and infinite space after $stage_n$, so that the real world input wait and real world output wait will never happen to $stage_1$ and $stage_n$, respectively. However, this assumption applies only to $stage_1$ and $stage_n$. Hence it is unlikely that the situations described by equations 7.33 or 7.35 are true for any of the other stages. Therefore, when compensating for the negative effects caused by the mismatching of iE_i_total_output_delay and iE_{i+1}_total_input_delay, it is beneficial to allocate more storage space to elements other than $element_1$ and $element_n$, since $\Sigma_\Delta E_1$_shifted$\hat{}_1$ and $\Sigma_\Delta E_n$_shifted$\hat{}_n$ are very likely to be greater than $\Sigma_\Delta E_i$_shifted$\hat{}_i$ (i=2..n-1). This is in agreement with the 'bowl phenomenon'.

7.4 CONCLUSION

In this chapter, the methodology and the results of chapter six which are based on the abstract representation of the system have been applied to examples which have practical purposes. The arguments used in the examples are based on the relationships between the causes and effects of production loss which were developed in chapter six. These examples showed that by representing a complex system with abstracted elements, a simpler system can be obtained and the relationships developed based on the simpler system are shown to be more intuitively praisable. These relationships are useful for providing insights about the system so that inferences about the system can be drawn.

The first example showed that if the cycle times in a balanced serial system obey the maximums calculated by the cycle time algorithm, then a certain production volume can be achieved in a given amount of time. The maximum cycle time algorithm has been summarized in a block diagram in figure 7.10. This algorithm limits certain cycle times

during the production and the limit is not less than (mct) minus the necessary transfer time. This could be a practical application if the production volume is the primary objective. Meeting this objective results in an overall lower mean cycle time for the stages in the system and a lower standard deviation in cycle times. However, using the definition of efficiency contained in chapter five, a higher efficiency resulted after the maximum cycle time algorithm was applied. Although the efficiency calculation takes the lowered mean cycle times into account, it ignores the lowering of the cycle time standard deviations. In other words, a higher efficiency in the results were obtained at the cost of the cycle time variation.

The expected production time in the maximum cycle time algorithm is defined to be the expected production output multiplied by the ideal world process time. The ideal world process time was originally designed to equal the mean cycle time of a balanced system. However, the maximum cycle time algorithm does not depend on the mean cycle time of the system. Therefore, it is possible to choose the ideal world processing time arbitrarily, as long as the ideal world processing times for all of the elements are equal. It is known that the system efficiency would be less than one when there are variations in the processing times. The ideal world processing time in the maximum cycle time algorithm can be chosen according to an expected maximum efficiency using equation 5.60, so that the numerator of equation 5.60 is determined by the system mean cycle time and the denominator is determined by the ideal world process time. The result of equation 5.60 is then the expected maximum system efficiency after the maximum cycle time algorithm is applied. It is believed that a lower expected system efficiency would lessen the decrease in the mean cycle times and the cycle time standard

deviations after the algorithm is applied.

The second example of optimal storage allocation was a simplified example for a system design stage application. Unlike many existing approaches, the example provided a method for storage allocation by using the long term individual behaviour measurements of each stage and by providing an objective for these measurements to fulfil using storage sizes as variables. The advantage of this approach is that it is theoretically independent of system size and is straightforward. However, this study provides only the preliminary theoretical grounds without any observable results. Details for fine tuning the approximate method are also open for future study. Nevertheless, this example presents how the methodology can be applied to other problems of the system.

Finally, the first example in this chapter was related directly to on-line system monitoring which is the motivation for the development of the methodology. The second example, although very simple, showed that it is possible to draw inferences about the system in general. It is hoped that the information provided can be useful to help solve different problems concerning the system which is the ultimate goal of the methodology.

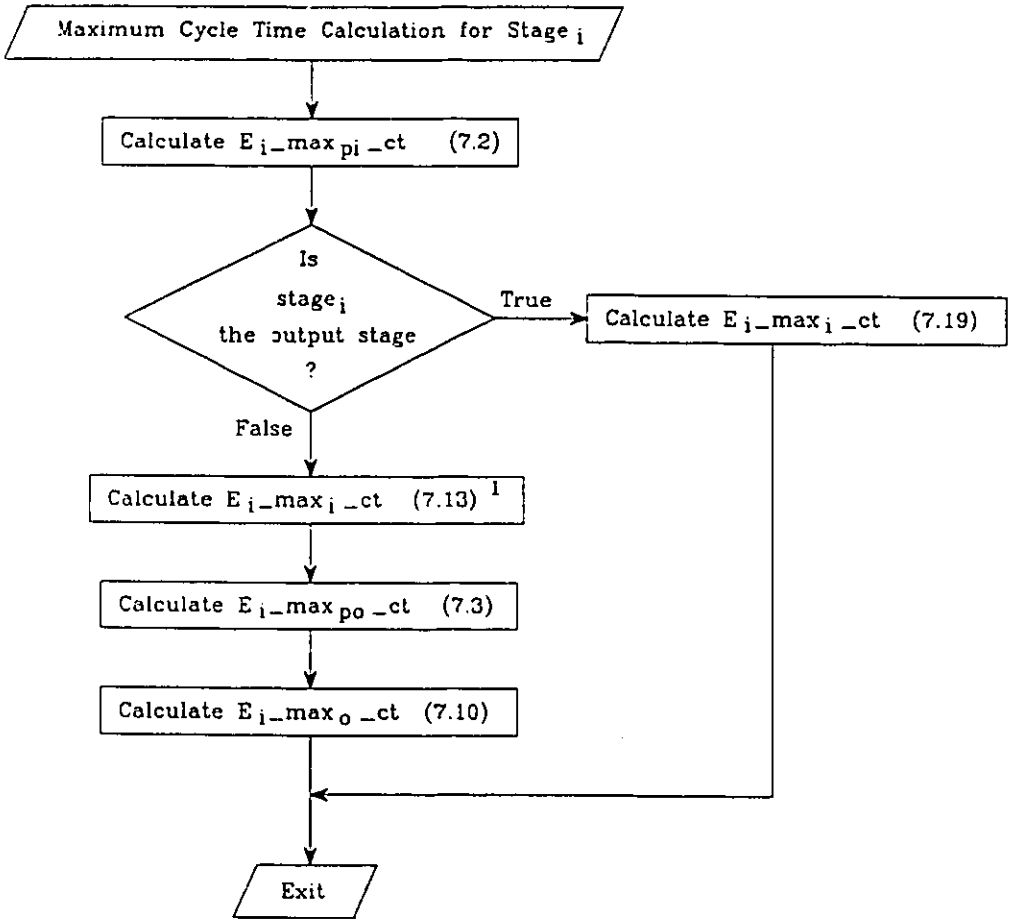


Figure 7.10 Block diagram for the maximum cycle time algorithm

Note 1: Equation 7.13 is rewritten for element$_i$

CHAPTER 8

PARALLEL CONFIGURATIONS

8.1 INTRODUCTION

In this chapter, the viability of applying the methodology to systems with parallel configurations is examined. The justification for having parallel configurations in a system can be found in (Buzacott 90) (Buxey et al. 72). In this study, parallel configurations are considered sub-sections of the system. Serial configurations make up the remaining sub-sections. The assumptions for the serial configuration sections remain the same as those previously stated. The assumptions for the parallel configuration sections are:

1. A joint which connects a serial section to a parallel section is a unique point. There is no parallel section which connects to another parallel section.
2. The mean cycle times of the stages in all of the lanes of a parallel section are equal.
3. The mean cycle time of a stage in a parallel section equals an integer multiple of the mean cycle time of a serial section. The value of this integer equals the number of lanes in the parallel section.

Each parallel section is connected to two serial sections by two joints, the splitting and merging joints. Figure 8.1 shows an example of a parallel section which has three

lanes.

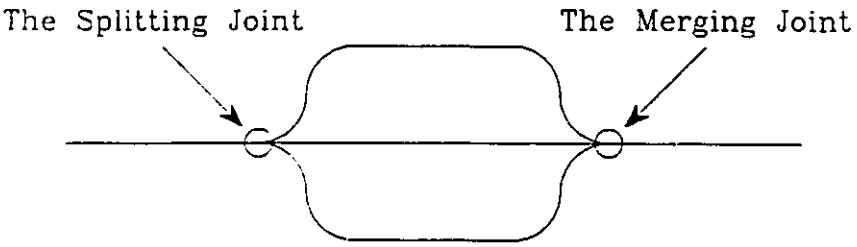


Figure 8.1 Two joints connecting a parallel section to two serial sections

There are two slightly different types of joints considered in this study which will be discussed in the following two sub-sections.

8.2 JOINT TYPE I

If a joint coincides with a stage in a serial section, it belongs to joint type I. This includes cases where a splitting joint is at, or immediately after the last stage of the serial section in front of it and a merging joint is at, or immediately before the first stage of the serial section following it. Using the previously defined symbols where a stage is represented by a square and a storage by a circle, the following diagram illustrates this type of joints.

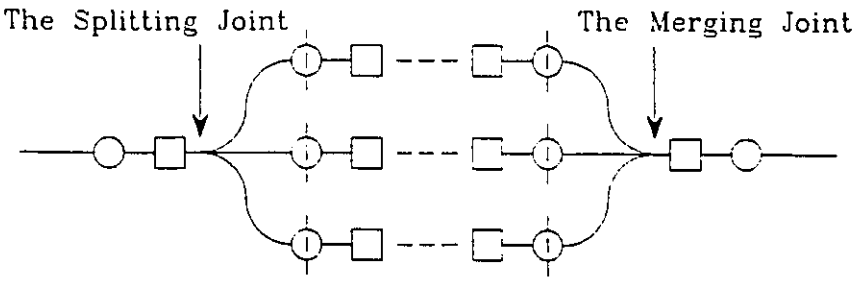


Figure 8.2 The type I splitting and merging joints

In this case, there is no storage in between the splitting joint and the last stage of the serial section in front of it, and in between the merging joint and the first stage of the serial section following it. Therefore, when forming the elements of the abstract representation, the storage at the beginning of a parallel section is divided up between the elements which include the leading stages of the parallel section and the element which includes the last stage of the serial section. The storage at the end of the parallel section is divided likewise between the elements which include the ending stages of the parallel section and the element which includes the first stage of the serial section. The locations of the divisions are illustrated by the dashed lines in figure 8.2. The configuration of figure 8.2 after representation by elements is shown in figure 8.3.

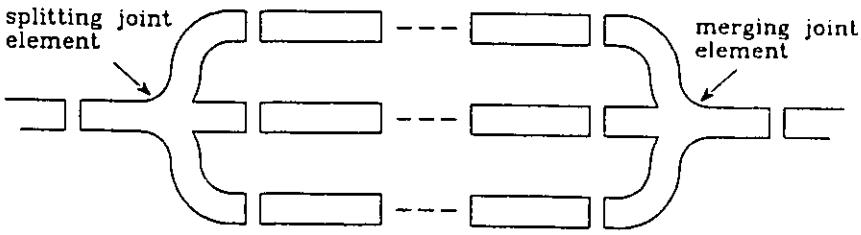


Figure 8.3 The abstract representation of the configuration shown in figure 8.2

In this case, the splitting and merging joints are embedded in the elements which include the last and first stage of the serial sections, respectively. It can be seen that each of the parallel lanes in between the splitting and merging elements forms a serial section by itself, with a slower mean cycle time than that of the serial sections of the system. Therefore, they are treated as serial sections. The elements, which include the splitting and the merging joints, are new ones introduced to support parallel configurations of this type. They are called the splitting element and the merging

element.

8.2.1 The Splitting Element Type I

A splitting element of this type has one input branch and several output branches, the number of which equals the number of lanes in the parallel section it belongs to. An element of a serial section can be treated as a special case of the splitting element where it has one input and one output branch. The variables of each output branch of a splitting element are the same as those of the output branch of a serial element. For example, there is a storage size limit and a current storage content for each individual branch. A superscript is used to identify the element variables for different branches. For instance, $iE_i_ob^1$ is the output storage content of branch 1 in element i in the ideal world. For any element, there is one 'ideal world input' for each input branch and one 'ideal world output' for each output branch. Therefore, the ideal world measurement of a splitting element can be measured individually for each of the output branches. The ideal world operation time of the element as a whole is unique, according to the principle of the conservation of time which is contained in chapter three. The splitting joint element from the 3-lane parallel section shown in figure 8.3 together with the 'ideal world input' and 'ideal world output' are shown in figure 8.4.

The constant processing time of the 'ideal world input' connected to a splitting element is the same as that of the serial section in front of it. The constant processing times of the 'ideal world outputs' are equal to the number of lanes multiplied by the ideal world processing time of the serial sections in the system. In the case of a 3-lane parallel section, the processing time of an 'ideal world output' is equal to (3 x mct).

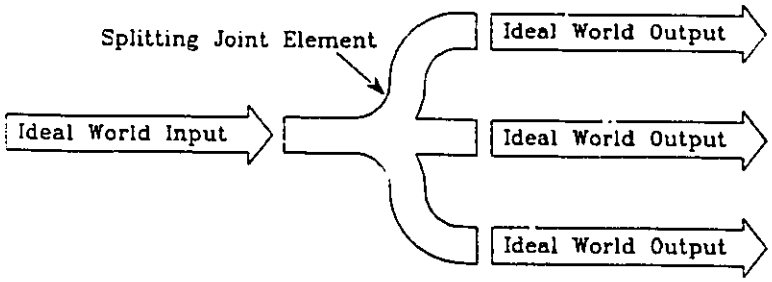


Figure 8.4 The ideal world input and ideal world outputs of a splitting element

8.2.2 Ideal World Measurement for Splitting Element Type I

Based on the ideal world measurement calculation for serial configurations, the ideal world measurement calculation for splitting elements of this type is straightforward. When calculating the ideal world measurement for splitting elements of this type, only one of the output branches and the 'ideal world output' connected to that branch needs to be considered at one time. This branch is the one selected to send the most currently finished job. The 'ideal world input' and the 'ideal world output' of the selected branch of the element form an ideal world measurement setup similar to that for a serial element. Therefore, the ideal world measurement for splitting elements of this type can be obtained in a way similar to that used for serial elements when the lane selection is known. Figure 8.5 illustrates this setup for the ideal world measurement of a splitting element with one of its output branches.

The ideal world measurement of the setup shown in figure 8.5 is calculated according to the calculation procedure for serial elements which is described in chapter four. The difference of the 'ideal world output' processing time in the parallel section and that in the serial section requires (mct) to be changed to (number of lane x mct) in

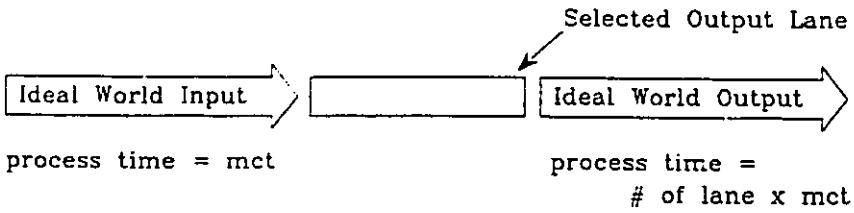


Figure 8.5 Ideal world measurement setup for a splitting element with a selected branch

the calculation of ideal world measurements involving the 'ideal world output'.

After the ideal world measurement setup of figure 8.5 has been formed using the selected branch and is updated normally according to the procedure described in chapter four, there is an extra step required to update (ob) of all of the branches other than the selected one. Defining M to be the number of lanes in a parallel section which has splitting element$_i$, j as the currently finished cycle and c as the chosen lane, the extra step for updating (ob) is described as follows:

$$\text{for } x = 1..M,\ x \neq c,$$
$$iE_i_ob^x_j = iE_i_ob^x_{j-1} - (M \times mct) \times (iE_i_Sct_j - iE_i_Sct_{j-1}) \qquad 8.1$$

This step keeps the output storage contents of all of the branches up to date because the ideal world update procedure, which was described in chapter four, only updates the output storage content of the selected output branch. The ideal world operation time which has elapsed from the start cycle time of the currently finished job to the start cycle time of the next job is unaccounted for in all of the other branches. Equation 8.1 decreases the output storage content of each of these branches according to the elapsed time and according to the rate of the 'ideal world output' operation. Any negative value for (ob) of any branch as a result of this operation will be taken care of

when a normal ideal world update is performed on that branch when the branch is chosen to send the finished job.

8.2.3 The Merging Element Type I

The merging element is the mirror image of the splitting element of the same type which has one output branch and several input branches. The number of input branches equals the number of lanes in the parallel section. The variables of each of the input branches of a merging element are the same as those of an input branch of a serial element. These variables are unique for each branch. The other properties of the merging element are also similar to those of the splitting element.

8.2.4 Ideal World Measurement for Merging Element Type I

The method of calculation used for the ideal world measurements of the merging element is similar to that used for the splitting element. Using the input branch from which the next job will come and the output branch of the element to form a setup similar to that of a serial element, the ideal world measurement can be calculated for this setup in a way similar to that used for a serial element.

In the calculation of the ideal world measurements involving the 'ideal world input', (mct) is changed to (number of lane x mct), which is the constant processing time of the 'ideal world input'. For the same reason as with the case of a splitting element, an extra step is needed to keep the input storage contents of all of the branches up to date. This extra step is applied after the ideal world status for the measurement setup using the selected input branch is updated normally. Using *c* as the lane from which the

next job will come, this step is described by:

$$\text{for } x = 1 \,..\, M,\ x \neq c,$$
$$iE_i_ib^x_j = iE_i_ib^x_{j-1} + (M \times mct) \times (iE_i_Sct_j - iE_i_Sct_{j-1}) \qquad 8.2$$

Equation 8.2 increases the input storage content of all of the branches except the selected branch, according to the rate of the 'ideal world input' operation for the ideal world operation time which has elapsed from the start cycle time of the most currently finished job to the start cycle time of the next job. Any negative value for (ib) of any branch as a result of this operation will be taken care of when a normal ideal world update is performed on that branch from which the next job comes.

8.3 JOINT TYPE II

If a joint coincides with a storage unit on a serial section, it belongs to joint type II. This includes cases where a splitting joint is at, or immediately after the last storage unit of the serial section in front of it and a merging joint is at, or immediately before the first storage unit of the serial section following it. Figure 8.6 illustrates this type of joints.

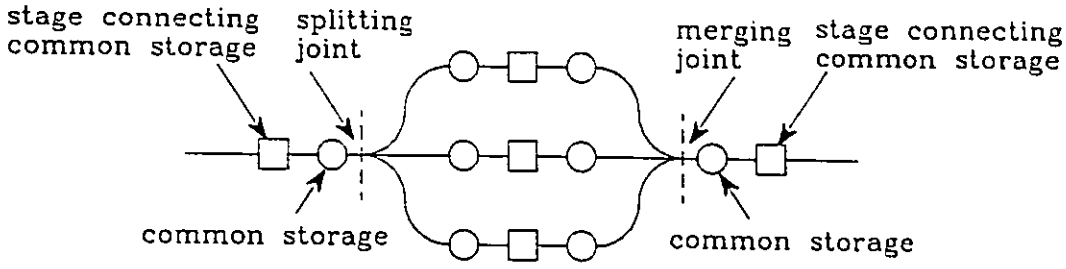


Figure 8.6 The type II splitting and merging joints

To include this type of joint in an element of the abstract representation is more complicated than to do so with the first type, because of the existence of storage between the joint and the stage in the serial section. The ideal world measurement setup using the selected branch which was applicable to the first type of joint, cannot be directly applied to this type of joint, because there should be a common area belonging to all of the branches of the element to implement the common storage between the joint and the serial section. For a splitting joint, the common storage is the storage in front of it and after the last stage of the serial section in front of it. For a merging joint, the common storage is the storage following it and in front of the first stage of the serial section following it. These storages are common to all of the branches of a joint of this type, since jobs sitting in this area can go to or come from any branch of the splitting or merging joint, respectively.

The common storage and the stage of the serial section connecting the common storage should ideally be represented by an element which should include a joint and several branches. However, the paths inside such an element which includes a joint will have two segments, one between the stage and the joint and another one between the joint and a branch. It will be very difficult to apply the ideal world measurement calculation

developed in chapter four to an element with paths of more than one segment. In order to avoid having any two segment paths in an element, a different form of the element will be considered.

The proposed elements for the splitting and the merging joints of this type are described in the following sections. The setup of this element form requires an extra restriction be applied when the storage is included into an element. This restriction requires all of the common storage spaces to be included in the element of the stage of the serial section connecting the common storage. The locations where the element boundaries must lay are represented by the dotted lines in figure 8.6. The abstract representation using elements for the configuration of figure 8.6 is shown in figure 8.7.

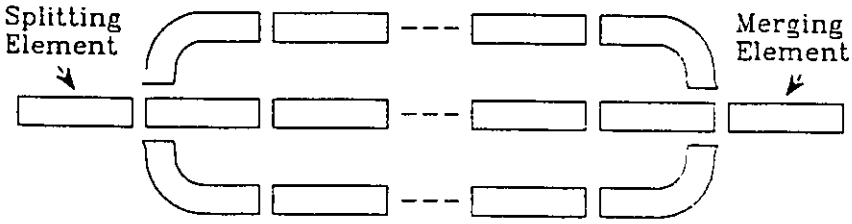


Figure 8.7 The abstract representation of the configuration shown in figure 8.6

8.3.1 The Splitting Element Type II

For the element setup, there is the restriction that requires the common storage at the end of the connecting serial section to be included in the splitting element. The splitting element should ideally look like this:

The ideal form of the splitting element includes a splitting joint at the output end, so that an output can go to one of the several branches when exiting the element and the

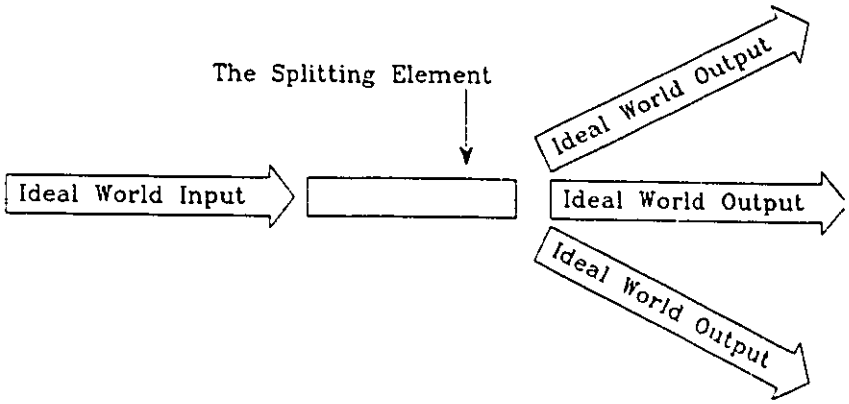


Figure 8.8 The ideal form of splitting element type II

path in between the joint and the output has zero length. There is a different 'ideal world output' for each branch used to measure each branch independently. The implementation of the ideal form of figure 8.8 is done by using the figure 8.4 configuration, which was originally used for the splitting element type I. The reason is that although the common storage is part of the serial section, the ideal world measurement for each branch of the element does not necessarily equal each other. Therefore, the figure 8.4 configuration is used to give each branch a unique set of variables so that each branch can be measured independently. The common storage from the serial section is implemented in all of the output branches using the element form of figure 8.4.

8.3.2 Ideal World Measurement for Splitting Element Type II

The difference in the ideal world measurement calculations between splitting elements of this type and that of type I is due to the existence of a common storage at the end of the serial section. For splitting elements type I, where there is no common

storage, the output branch selection for the current job from the last stage of the serial section must be known before the next job can leave the stage. The existence of a common storage creates a gap in time, from the time the stage, at the end of the serial section, finishes a job to the time the output branch selection is known for the job. Therefore, the ideal world calculation for this type of splitting element is not as straightforward.

There are two extra situations to be considered in the ideal world calculation of splitting elements of this type. The first situation occurs when the job, which is currently being finished by the stage at the end of the serial section, stays in the common storage after the next job is finished. The second situation occurs when the splitting joint sends out a job from the common storage and the stage has not recently finished a job. These two situations will be expanded upon later. There is a normal situation similar to that of splitting element type I, which occurs when the job which has currently been finished by the stage is sent out to a branch before the next job is finished by the stage. In summary, there are three situations in the ideal world calculation of splitting elements of this type. When the first situation arises, the storage contents of all of the output branches of the element which are in the form shown in figure 8.4 should be increased. When the second situation arises, the storage contents of all of the output branches of the element should be decreased. When the normal situation arises, the method described for splitting element type I is used.

For the first situation, the ideal world calculation for the input branch of the element is performed as described in chapter 4. For the output branches of the element, the following equation is used to update the output storages:

for $x = 1 .. M$,

$$iE_{i_}ob^{x}_{j} = iE_{i_}ob^{x}_{j-1} + 1 - (M \times mct) \times (iE_{i_}Sct_{j} - iE_{i_}Sct_{j-1}) \qquad 8.3$$

This operation adds the currently finished job to the common storage image in each branch, and keeps track of the effect of the 'ideal world output' on each branch for the elapsed ideal world operation time. $iE_{i_}Sct_{j}$ is updated as described in section 4.4 by letting $iE_{i_}$output_wait be zero. The upper and lower limits of the output storage of a branch and the effect of exceeding any one of them are considered when the normal update of the branch is performed.

For the second situation, an ideal world update, as described in chapter four, is not necessary since there is no job being finished. The following operation is performed for all of the branches:

$$\text{for } x = M, \quad iE_{i_}ob^{x}_{j} \leftarrow iE_{i_}ob^{x}_{j} - 1 \qquad 8.4$$

The above operation reduces one job from the common storage. The 'ideal world output' has no effect on the output branches since no ideal world operation time has elapsed. The effects the output storage limits have are handled in the same way it was handled in the first situation.

The implementation of the element tries to use as many normal updates as possible by using a single item queue. Assuming that the event of the stage finishing a job with cycle time (ct) comes as an end of cycle message (EOC ct), and that the event of the splitting joint sending out a job to branch x comes as a branching message (BRANCHING x), the algorithms are as follows:

```
EOC( Ct ):
    Previous_Ct ← qOut();
    qIn( Ct );
    IF          Previous_Ct exist
    THEN        Perform the procedure of situation 1
    ELSE        End
```

Text Box 7.1 Algorithm for an End of Cycle event

```
BRANCHING( x ):
    Current_Ct ← qOut();
    IF          Current_Ct exist
    THEN        perform normal ideal world update using Current_Ct on
                branch x
    ELSE        Perform procedure of situation 2
```

Text Box 7.2 Algorithm for a Branching event

where qIn(v) puts v into the single item queue, and qOut() takes out and returns with the content of the single item queue.

8.3.3 The Merging Element Type II

The ideal form of the merging element of this type is shown in figure 8.9. The ideal form of the merging element includes a merging joint at the input end, so that an

input can come from one of the several branches when entering the element and the path in between the input and the joint has zero length. A different 'ideal world input' for each input branch is used to measure each branch independently. However, using the same argument as was used with the case of the splitting element of the same type, the configuration of merging element type I is used to implement the merging element of this type.

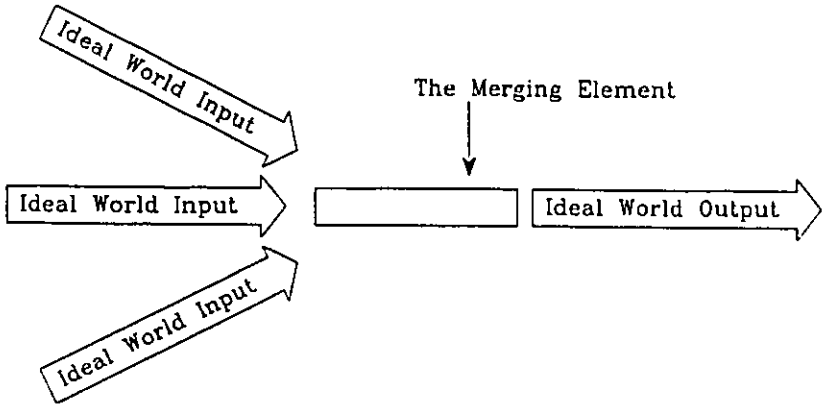


Figure 8.9 The ideal form of merging element type II

8.3.4 Ideal World Measurement for Merging Element Type II

The ideal world calculation for the merging element of this type is a mirror image of that of the splitting element of the same type. There are also two extra situations in the ideal world measurement calculation. The first situation occurs when the stage takes a job from the common storage and no job has come to the common storage from the end of the previous cycle to the end of the current cycle. The second situation occurs when the merging joint sends a job into the common storage and the stage has not recently finished a job. The normal situation occurs when the merging joint sends a job into the

common storage after the stage has recently finished the current job and before finishing the next job.

When the first situation arises, the storage contents of all of the input branches of the element in the form of merging element type I should be decreased. When the second situation arises, the storage contents of all of the input branches of the element should be increased. When the normal situation arises, the method described for merging element type I is used.

For the first situation, the ideal world calculation for the output branch of the element is performed as described in chapter 4. For the input part of the element, the following update is performed on the input storages:

$$\text{for } x = 1 \,..\, M,$$
$$iE_{i_}ib^{x}_{j} = iE_{i_}ib^{x}_{j-1} - 1 + (M \times mct) \times (iE_{i_}Sct_{j} - iE_{i_}Sct_{j-1}) \qquad 8.5$$

For the second situation, the following operation alone is performed on all of the input branches:

$$\text{for } x = 1..M, \quad iE_{i_}ib^{x}_{j} \leftarrow iE_{i_}ib^{x}_{j} + 1 \qquad 8.6$$

Assuming that the event of the merging joint sending a job into branch x comes as a branching message (BRANCHING x), the algorithms defined in the case of splitting joints of this type are already applicable to the merging joint implementation, using the procedures described for situations 1 and 2 in this section.

8.4 REAL WORLD PARAMETERS CALCULATION FOR PARALLEL CONFIGURATIONS

For the methodology to be useful to parallel configurations, the calculation of the real world parameters based on ideal world measurements is necessary. The real world parameters calculation for serial configurations has been described in chapter five. This section briefly discusses some of the issues concerning the real world parameters calculation for parallel configurations.

After the ideal world measurements calculation for splitting and merging elements is established, the calculation for real world parameters should ideally follow that which is described in chapter five. Calculation of the real world parameters between a splitting or merging element and an element from the serial section or an element from a parallel section should be similar to that between any two neighbouring element in a serial configuration, except the mean cycle time (mct) is changed to (number of lane x mct) when an element from a parallel section is involved.

The situation is ideal for the case of joint type I so that the real world parameter calculations between a splitting or merging element and a serial or parallel element are similar to those between any two neighbouring elements in a serial configuration. However, the real world parameters calculation involving joint type II is a little more complicated and requires some extra concern. Those details though are not discussed here. Examples involving both types of joints were tested and the results were satisfactory for all of the test cases since the calculated real world parameters exactly matched those of the emulator output.

8.4.1 Blocking Between Parallel Lanes

There is one point worth noting regarding the differences between the real world parameter calculations for the two types of joints. The stage and the joint are not separable for elements of joint type I but this is not true for elements of joint type II. This difference is made obvious by the way a parallel lane is blocked before a merging joint. In the case of joint type I, a parallel lane can only be blocked by the merging element either directly or indirectly. In the case of joint type II, a parallel lane can be blocked by the merging element or by other parallel lanes. The blocking of a parallel lane by another parallel lane at the merging joint can occur if the sequence of jobs exiting the parallel section has to follow a certain order. For example, if lane 1 is full from the exiting stage to the merging joint, and the storage at the merging joint is empty but is not available to lane 1, because the job being currently processed by lane 2 must proceed the jobs in lane 1 when exiting the parallel section, then lane 1 is blocked by lane 2. This is illustrated by figure 8.10.

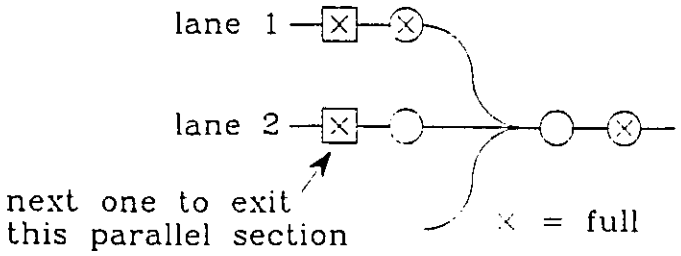


Figure 8.10 Example of a parallel lane being blocked by another parallel lane

This situation is not obvious in the case of joint type I because joint type I coincides with a stage instead of a storage as shown in figure 8.10. Then if the same situation of lane 1 being blocked by lane 2 arises, it will first be treated by the merging

element as an input wait between the merging element and the last stage of lane 2. The merging element is then 'shifted' by the last element of lane 2 and this 'shifted' amount in turn contributes to the blocking of the last element in lane 1 by the merging element. On the other hand, in the case of joint type II in the same situation, an input wait between the last stage of lane 2 and the merging element does not exist since the storage between them is not empty. Therefore, an extra step is required to calculate the amount by which lane 2 has blocked lane 1. This step assumes there is a direct connection from lane 1 to lane 2 as shown in figure 8.11.

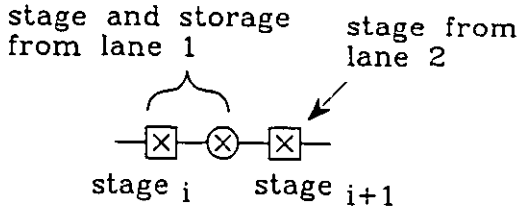


Figure 8.11 An hypothetical connection between two parallel lanes

However, the condition for a real world output wait between $stage_i$ and $stage_{i+1}$ as defined in chapter five does not exactly hold true in the case of an hypothetical connection. This is because equation 5.8, which is reproduced here, no longer holds true:

$$N_i \leq N_{i+1} + S_i \qquad 8.7$$

where N_i is the number of jobs finished by $stage_i$ since the beginning. S_i is the number of initial empty storage spaces in the real world between $stage_i$ and $stage_{i+1}$. Equation 5.8 does not hold true in this case because the jobs processed by $stage_{i+1}$ are not supplied by $stage_i$. Therefore, equation 5.8 has been modified for the derivation of the condition

for the output wait between parallel lanes as follows:

$$N_i \leq N_{i+1} + S_i + D \qquad 8.8$$

In this case, S_i is the initial empty space in the real world output storage between $stage_i$, which is the last stage of a lane, and the merging joint. D is an integer which directly relates to the difference in the number of jobs distributed to the two lanes containing $stage_i$ and $stage_{i+1}$ respectively, by the splitting joint of the parallel section since the beginning. (D x mean cycle time of $stage_i$ or $stage_{i+1}$) in turn contributes directly to the shifting of $element_i$ after it is blocked by $element_{i+1}$ of another parallel lane, in a parallel section which has a merging element type II.

In summary, if the sequence of jobs exiting a parallel section have to follow a certain order, real world output waits between two parallel lanes are inevitable if the merging element is of type II. The condition under which a real world output wait between two parallel lanes occurs is stated in equation 8.8, which is a modification of equation 5.8 by the introduction of an integer 'D'. The effect of shifting the blocked element contributed by 'D' in equation 8.8 is valuable to the study of lane selection decisions made by the splitting joint of the same parallel section. The objective of this study would be to base the lane selection decision of a splitting joint on the minimization of the negative 'shifted' effect contributed by 'D' of equation 8.8.

8.5 SOFTWARE IMPLEMENTATION CONSIDERATIONS

The class hierarchy which was implemented using the C++ language for exploratory programming discussed in section 6.6 is modified slightly in order to support

the parallel configurations discussed in this chapter. The configuration shown in figure 6.8 is ideal for the ideal world class of an element of the serial section, which has one input and one output branch. For an element including a joint for a parallel section, multiple 'input sub-classes' or 'output sub-classes' are required. However, the dynamic multiple inheritance of a same sub-class cannot be implemented. Therefore, the implementation of the dynamic multiple inheritance of a sub-class is done by implementing the 'input' and 'output' sub-classes as two individual classes and they are the 'friends' of the 'ideal world' class. Two arrays of pointers to objects belonging to the 'input' and 'output' classes are kept in the ideal world class.

8.6 CONCLUSION

This chapter briefly discussed the procedures involved with the application of the methodology to systems with parallel configurations. The parallel configurations considered in this study include two types of joints. Each joint can be a splitting or merging joint. The parallel sections were then connected to the serial sections of the system through these joints, according to the assumptions stated at the beginning of the chapter. The discussion concentrated on how to integrate the elements which contained any of these joints into the methodology. Each parallel lane between a splitting element and a merging element was then treated as a serial configuration with a different mean cycle time.

For elements containing the first type of joints, the procedure used to obtain the ideal world measurement and to construct the real world parameters only required slight modifications of the existing one for serial configurations. For elements containing the

second type of joints, two extra situations were required. These two situations were handled by two event-driven subroutines which were shown in text boxes 8.1 and 8.2. Moreover, for parallel sections which have a second type merging element, blocking between parallel lanes is possible if jobs exiting the merging joint must follow a certain sequence. This type of blocking is detected by the condition shown in equation 8.8.

Systems with parallel configurations of both types were tested with the assumption that jobs exiting a merging element type II will follow the same sequence as the one followed when coming into the parallel section. The lane selection decision of a splitting joint is a simple one which is based on the current status of all of the lanes in a parallel section. Numerous cases were used to test these systems, each requiring 20,000 jobs to be produced. The cycle time density function of figure 7.1 is used by all of the stages. The results were satisfactory for all of the test cases and the calculated real world parameters exactly matched those of the emulator output. However, an analysis and proof of the procedures used to obtain the ideal world measurements and to construct the real world parameters should be the next step in the study of parallel configurations.

CHAPTER 9

CONCLUSION

9.1 SUMMARY

Assembly systems with inter-stage storages are an important class of systems in many aspects including manufacturing (Gershwin 83). The existing studies concerning this type of system have generally concentrated on probabilistic approaches, the first of which was performed by (Vladzievskii 52) according to (Gershwin 83). The study in (Masso et al. 74) stated that, the existing analytical techniques are too complex and cumbersome and it is believed that simulation techniques do not give much ground for drawing inferences in analysing the production system. Most of the existing approaches are also severely limited by the system size.

The current study is an attempt to provide a new point of view to the study of systems with inter-stage storages. The approach taken by the current study is based on the instantaneous relationships of a system through the simplification provided by the use of an abstract representation. The author believes that the instantaneous relationships of a system is the missing link which can provide information that has been left out by the existing studies. Moreover, the approach is not limited by system size and potentially supports the implementation of parallel configurations.

The current approach focuses on representing and quantitatively 'explaining' the

present state of a system instead of devising a model to 'predict' system behaviour as is done in many of the existing approaches. The approach in this study emphasizes the measurement of the local effects of a stage instead of assuming the cycle time distributions as has been done in many studies and has been shown to be unrealistic. The derivation of the current approach although mathematically elementary, is a process involving hypothesis, analysis and reasoning. The simplicity of the results as a consequence encourages understanding and allows for the inference of knowledge about the system. Therefore, the information provided by the approach is simple and straightforward and highlights insights about the operations of the system, which may otherwise be obscured by the complicated nature of the system. Moreover, the procedures in the approach are ready to be implemented as a real time system to produce results along with real productions, since all calculations involved are deterministic.

9.2 DISCUSSION

The abstract representation, which simplifies the system being analyzed and allows for direct inferences to be drawn as demonstrated by the two examples in chapter seven, is the foundation of the current study. The abstract representation allows a n-stage system with inter-stage storages to be represented by a n-element system. The input and output boundaries of an element can be arbitrarily chosen as long as the rules defining an element are satisfied. By inspecting the 'ideal world' measurement derivation and the calculation of the real world parameters which are based on the measurement results, it is obvious that it is beneficial for an element to have more storage space. This can be observed in the process suggested for on-line system monitoring that, for a system

with identical stages (equal mean and deviation in cycle times) the more storage space an element has, the less likely the element would cause production loss at the output of the system. Moreover, in the 'maximum cycle time' example, an element which has more storage space than the others, will have less restrictive maximum cycle time limits than those of the other elements. The results of these two applications depends on the choice of the element boundaries. Therefore, fairness among elements must be considered when choosing element boundaries for certain types of applications.

In a system with similar stages (equal mean and deviation in cycle times) the element boundaries can be chosen so that the storage space between any two neighbouring stages is divided up equally between the two elements. However, in a system with different stages (cycle time deviations among the stages are not equal), the criterion recommended for 'storage allocation' should be considered. In short, this criterion is that the long term ideal world total output delay of any element should match the long term ideal world total input delay of its immediate downstream element.

The fairness in element setup affects the fairness of the results in some applications. On the other hand, the 'storage allocation' example in chapter seven employed the advantage of the arbitrary choice of element boundaries, so that storage allocation can be performed on serial systems virtually independent of system size.

9.3 CONTRIBUTION AND CONCLUSION

The contributions of this study which flow from the attempts to satisfy the objectives stated in chapter three are as follows:

1. An abstract representation of the system has been devised. The independent

measurement for each element of the abstract representation has been defined. An exact reading of the defined independent measurement has been formulated for an element. Calculations of the exact values of the real world parameters have been formulated based on the independent measurement results. The instantaneous cause and effect relationships for each element and the output of the system have been formulated. Therefore, the first objective has been satisfied.

2. It was argued that using the instantaneous cause and effect relationships, the origins of production loss can be quantitatively shown at the system output and satisfied the purpose of on-line system monitoring as defined. Thus the second objective has also been satisfied.

3. It has been shown through the 'maximum cycle time' example that the instantaneous cause and effect relationships are useful for drawing inferences about the system. Furthermore, the theoretical example of 'storage allocation' suggested an approximate procedure which is virtually independent of system size. This example showed that the instantaneous cause and effect relationships are not limited to only real time analysis applications. Thus a step towards satisfying the third objective has been made.

By satisfying the first objective, an abstract representation of the system is formed which precisely describes the current system state according to the assumptions. In fact, only a slight modification such as the adjustment of the assumed constant transfer time to the real value at each end of cycle is required for this approach of system monitoring to be implemented and incorporated into the existing system monitoring procedures, which usually collects cycle time durations for statistical analysis.

The calculation of the maximum cycle time limits using the procedure as

described in the 'maximum cycle time' example can be very useful in preventing system production loss. Furthermore, this information can be used in other areas concerning the system such as scheduling jobs which require different cycle times for the system.

9.4 RECOMMENDATIONS FOR FUTURE RESEARCH

It is hoped that subsequent studies can be performed based on the results of the current study, to utilize the advantage of the directness of the current approach. These future studies can focus on improving the results and validating the assumptions made in this study. Furthermore, it would be interesting to apply this approach to other problems concerning the same type of system, so that the optimal goal of making the current approach one of the general approaches used to study production systems with inter-stage storages can be reached.

APPENDIX A

System Output 'Shifted' Plots for a ten-stage system

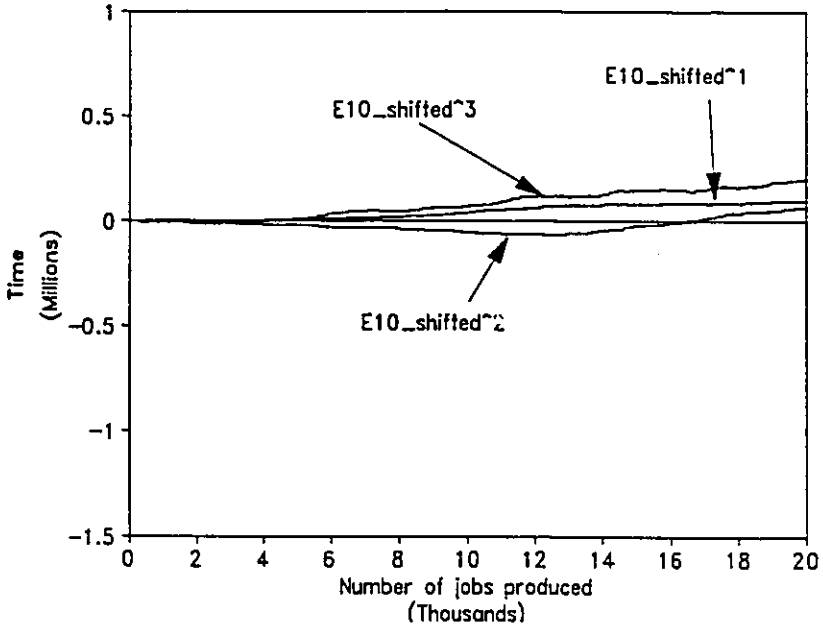


Figure A.1 E_{10}_shifted^$_i$ i=1..3, vs. number of jobs before applying the MCT algorithm

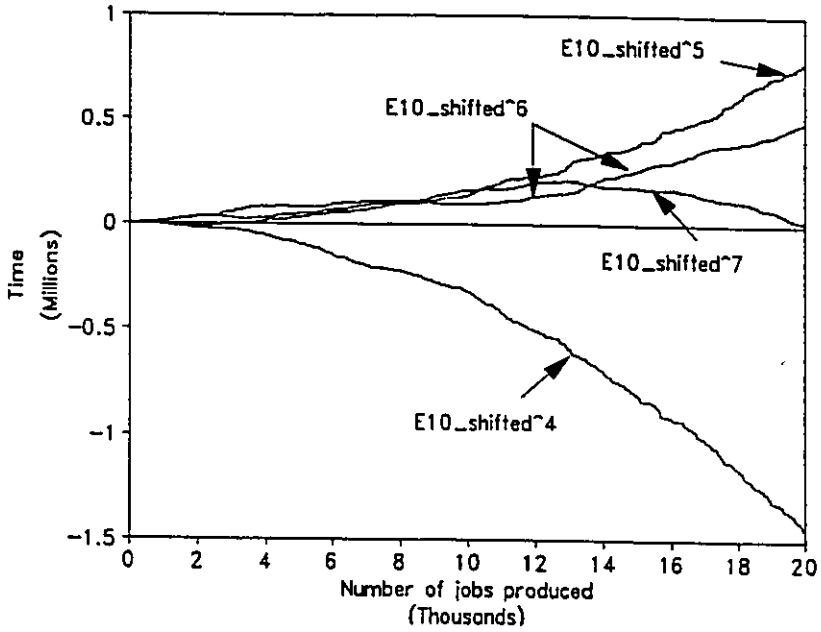


Figure A.2 E_{10}_shifted^$_i$ i=4..7, vs. number of jobs before applying the MCT algorithm

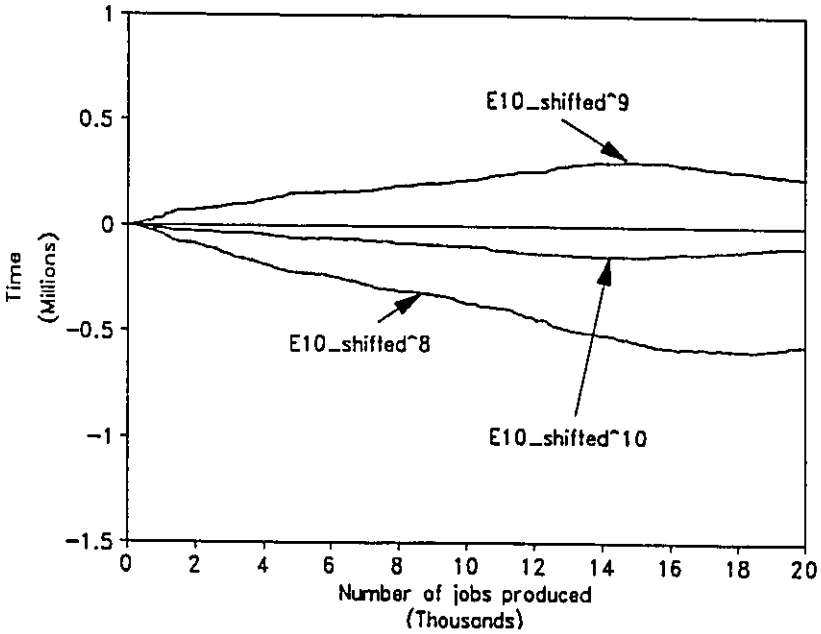


Figure A.3 E_{10}_shifted^$_i$ i=8..10, vs. number of jobs before applying the MCT algorithm

	Mean	Std. dev.	Minimum	Maximum
$Stage_1$	59.9	29.9	1.5	294.3
$Stage_2$	60.0	29.8	1.5	250.8
$Stage_3$	59.9	29.9	2.5	294.5
$Stage_4$	60.2	30.2	3.2	240.0
$Stage_5$	60.0	30.0	2.1	316.7
$Stage_6$	59.9	29.6	1.7	237.4
$Stage_7$	60.0	29.9	2.8	274.5
$Stage_8$	60.0	30.1	2.1	251.1
$Stage_9$	60.0	30.1	2.2	273.7
$Stage_{10}$	59.9	29.8	3.6	267.3
Average	60.0	29.9	2.3	270.0

Table A.1 Cycle time statistics before applying the MCT algorithm

Actual production time = 1546685.1

Error (percentage of expected production time) = -28.89%

Efficiency = 77.56%

APPENDIX B

System Output 'Shifted' Plots after applying the maximum cycle time 'algorithm'

MF = 1.00

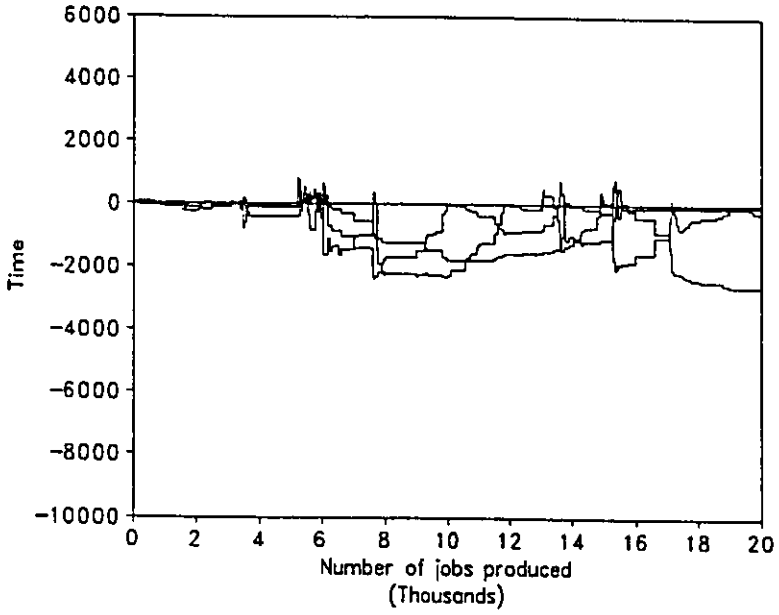


Figure B.1 E_{10}_shifted$_i$, i=1..3, vs. number of jobs after applying the MCT algorithm
MF = 1.00

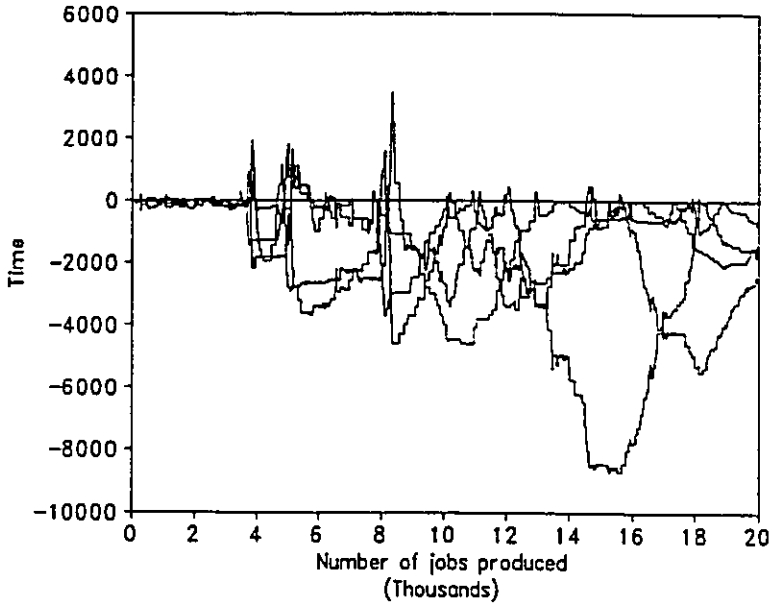


Figure B.2 E_{10}_shifted^$_i$ i=4..7, vs. number of jobs after applying the MCT algorithm MF = 1.00

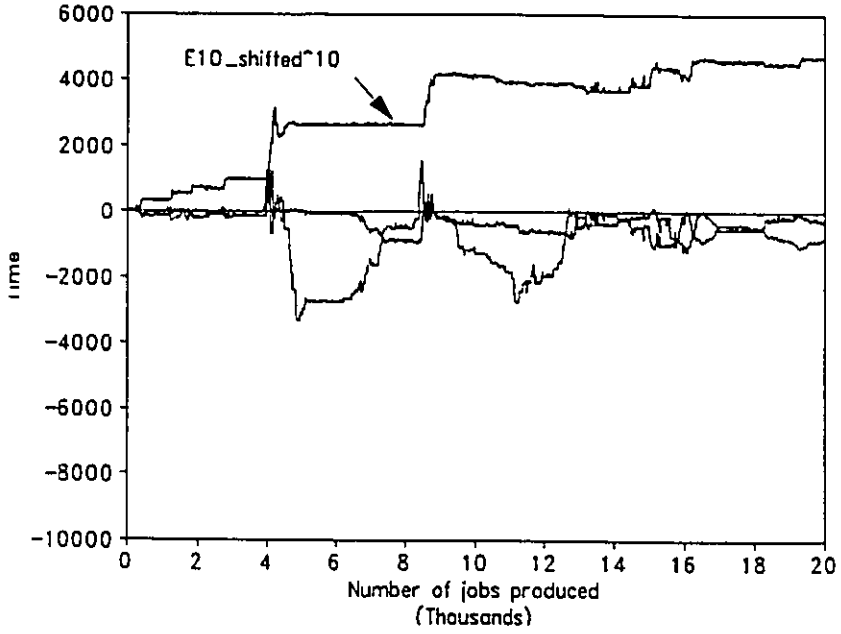


Figure B.3 E_{10}_shifted^$_i$ i=8..10, vs. number of jobs after applying the MCT algorithm MF = 1.00

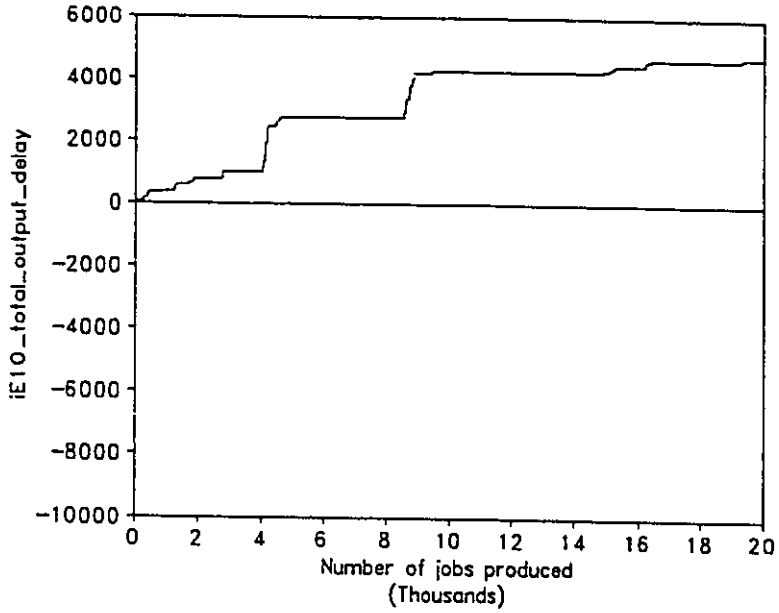


Figure B.4 iE_{10}_total_output_delay vs. number of jobs after applying the MCT algorithm MF = 1.00

	Mean	Std. dev.	Minimum	Maximum
$Stage_1$	53.45	20.36	1.51	206.12
$Stage_2$	53.51	20.39	1.45	177.97
$Stage_3$	53.35	20.29	2.47	194.03
$Stage_4$	53.52	20.41	3.19	156.22
$Stage_5$	53.52	20.38	2.13	201.06
$Stage_6$	53.52	20.35	1.71	190.26
$Stage_7$	53.51	20.36	2.82	181.19
$Stage_8$	53.46	20.49	2.06	191.16
$Stage_9$	53.44	20.40	2.24	178.16
$Stage_{10}$	53.43	20.30	3.58	185.74
Average	53.47	20.37	2.32	186.19

Table B.1 Cycle time statistics after applying the MCT algorithm, MF = 1.00

Actual production time = 1210403.3

Error (percentage of expected production time) = -0.87%

Efficiency = 88.35%

APPENDIX C

System Output 'Shifted' Plots after applying the maximum cycle time 'algorithm'

MF = 0.75

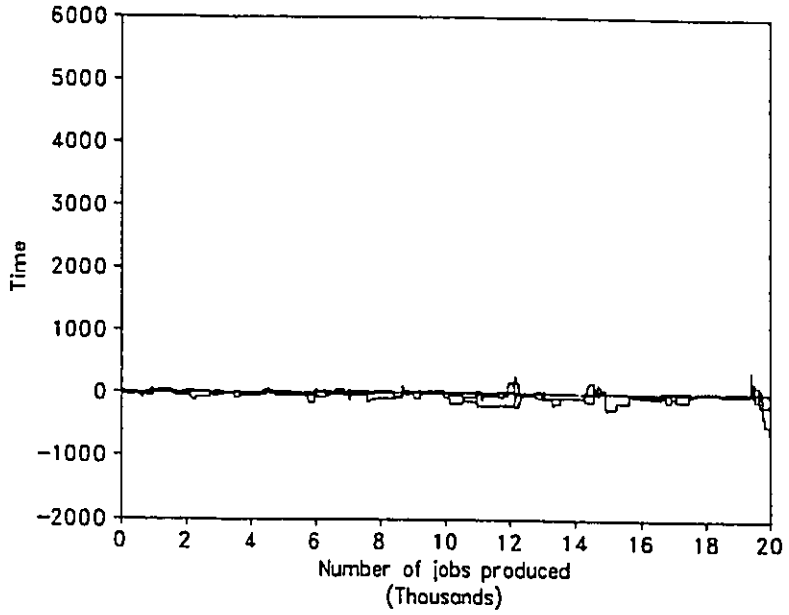


Figure C.1 E_{10}_shifted^$_i$ i=1..3, vs. number of jobs after applying the MCT algorithm
MF = 0.75

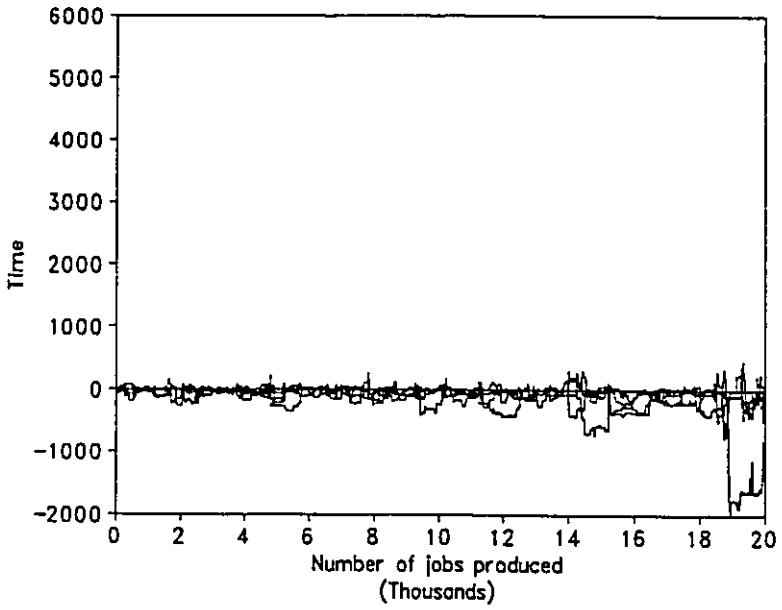


Figure C.2 $E_{10}_shifted\hat{}_i$ i=4..7, vs. number of jobs after applying the MCT algorithm MF = 0.75

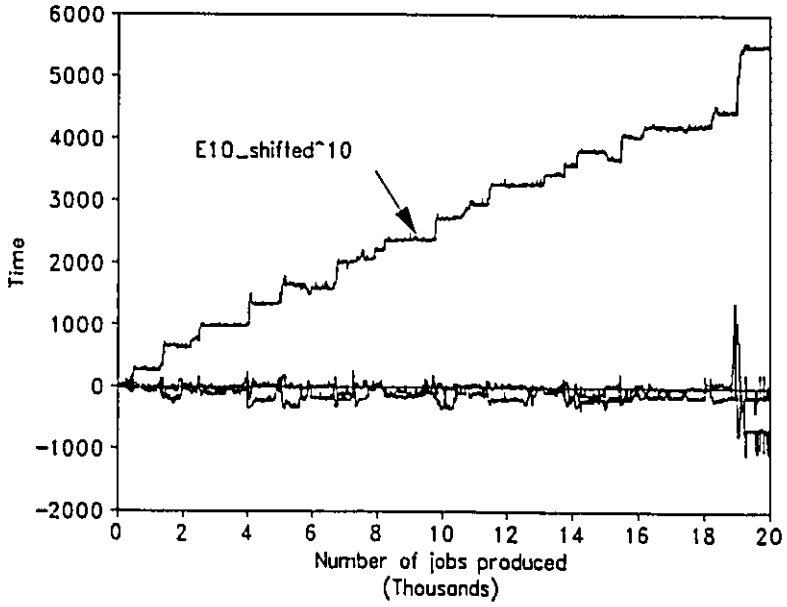


Figure C.3 $E_{10}_shifted\hat{}_i$ i=8..10, vs. number of jobs after applying the MCT algorithm MF = 0.75

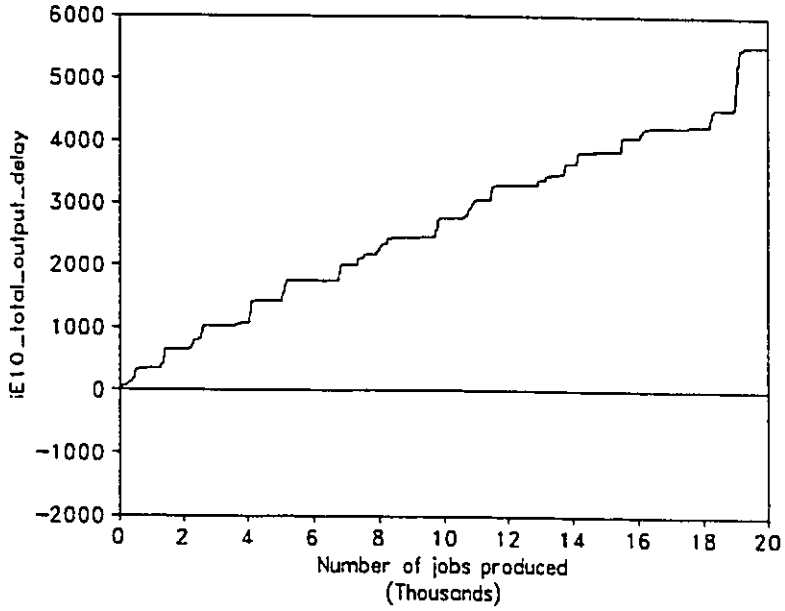


Figure C.4 iE_{10}_total_output_delay vs. number of jobs after applying the MCT algorithm MF = 0.75

	Mean	Std. dev.	Minimum	Maximum
$Stage_1$	53.48	20.36	1.51	157.82
$Stage_2$	53.54	20.40	1.45	149.48
$Stage_3$	53.37	20.27	2.47	154.06
$Stage_4$	53.54	20.44	3.19	156.76
$Stage_5$	53.54	20.43	2.13	171.30
$Stage_6$	53.55	20.36	1.71	173.28
$Stage_7$	53.51	20.37	2.82	172.83
$Stage_8$	53.48	20.51	2.06	202.25
$Stage_9$	53.48	20.45	2.24	213.75
$Stage_{10}$	53.47	20.35	3.58	163.98
Average	53.50	20.39	2.32	171.55

Table C.1 Cycle time statistics after applying the MCT algorithm, MF = 0.75

Actual Production Time = 1203353.5

Error (percentage of expected production time) = -0.27%

Efficiency = 88.91%

APPENDIX D

System Output 'Shifted' Plots after applying the maximum cycle time 'algorithm'

MF = 0.5

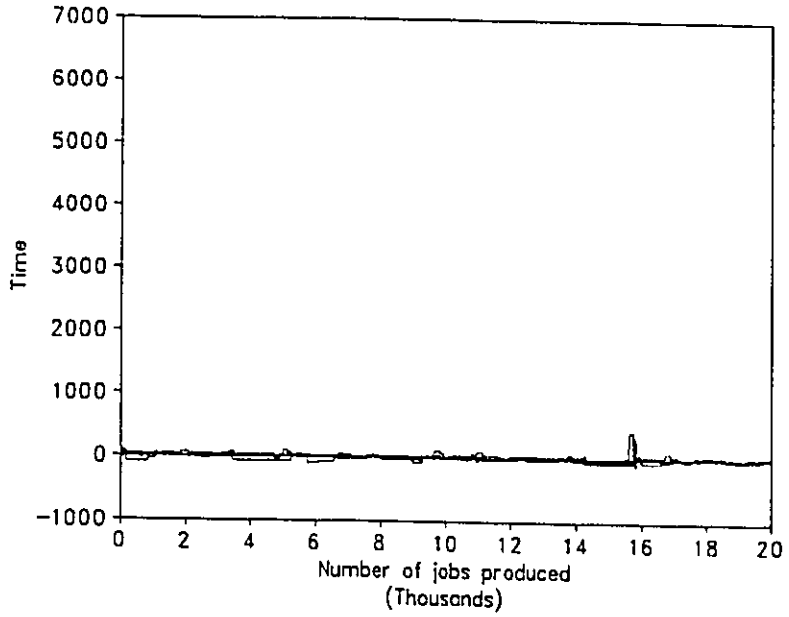


Figure D.1 E_{10}_shifted^$_i$ i=1..3, vs. number of jobs after applying the MCT algorithm MF = 0.50

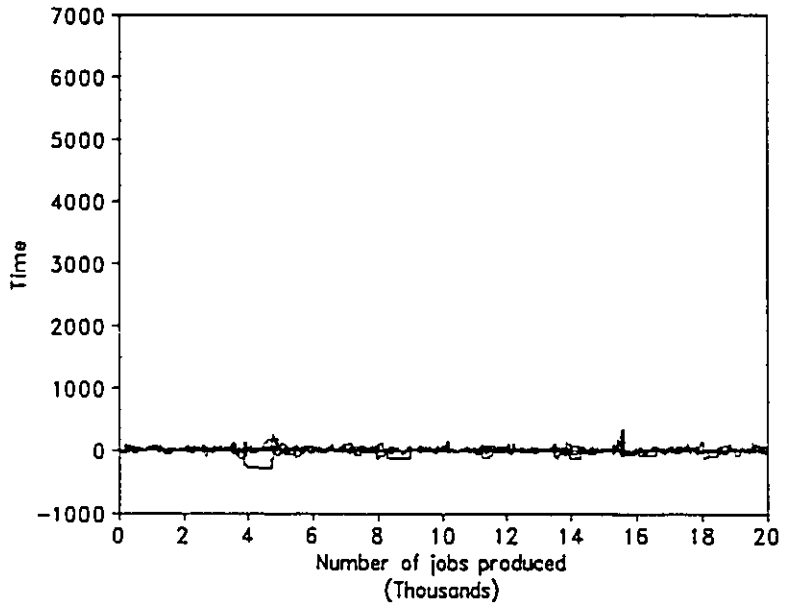


Figure D.2 E_{10}_shifted^$_i$ i=4..7, vs. number of jobs after applying the MCT algorithm MF = 0.50

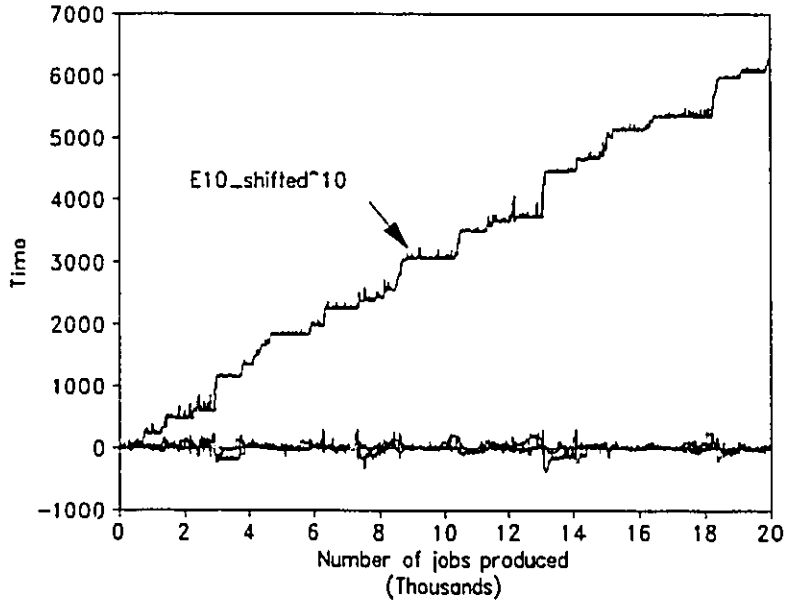


Figure D.3 E_{10}_shifted^$_i$ i=8..10, vs. number of jobs after applying the MCT algorithm MF = 0.50

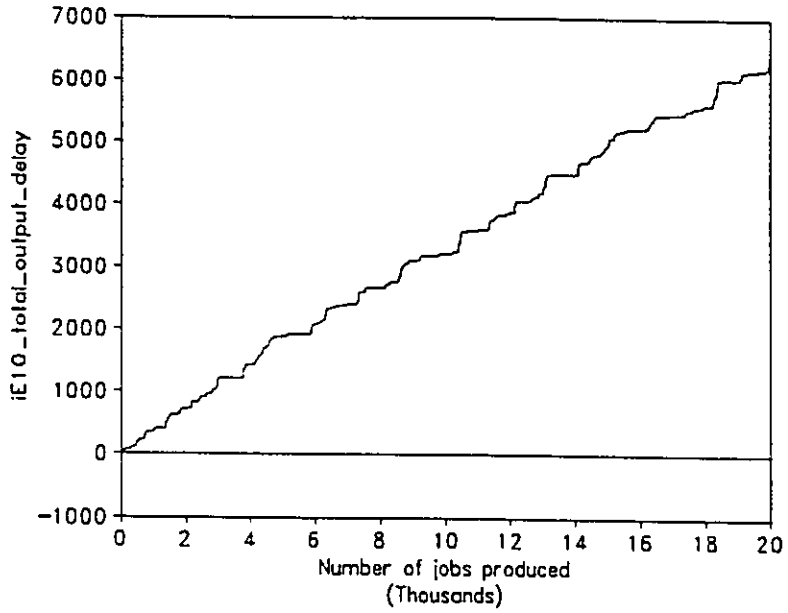


Figure D.4 iE_{10}_total_output_delay vs. number of jobs after applying the MCT algorithm MF = 0.50

	Mean	Std. dev.	Minimum	Maximum
$Stage_1$	53.54	20.38	1.51	151.11
$Stage_2$	53.60	20.46	1.45	125.05
$Stage_3$	53.42	20.31	2.47	142.92
$Stage_4$	53.60	20.48	3.19	127.93
$Stage_5$	53.59	20.45	2.13	129.49
$Stage_6$	53.60	20.44	1.71	146.32
$Stage_7$	53.58	20.42	2.82	130.50
$Stage_8$	53.54	20.53	2.06	141.91
$Stage_9$	53.52	20.45	2.24	157.64
$Stage_{10}$	53.51	20.40	3.58	180.02
Average	53.55	20.43	2.32	143.29

Table D.1 Cycle time statistics after applying the MCT algorithm, MF = 0.50

Actual Production Time = 1199965.4

Error (percentage of expected production time) = 0.0029%

Efficiency = 89.25%

APPENDIX E

System Output 'Shifted' Plots after applying the maximum cycle time 'algorithm'

MF = 0.25

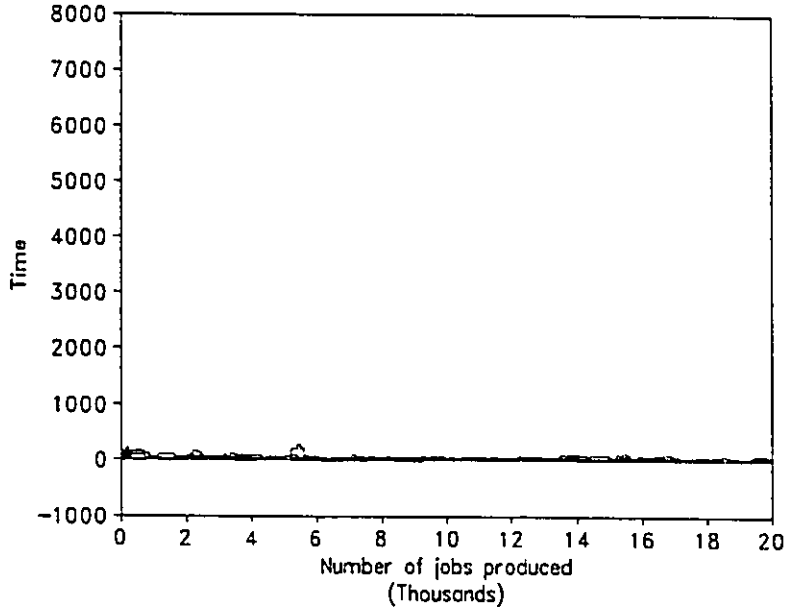


Figure E.1 E_{10}_shifted^$_i$ i=1..3, vs. number of jobs after applying the MCT algorithm MF = 0.25

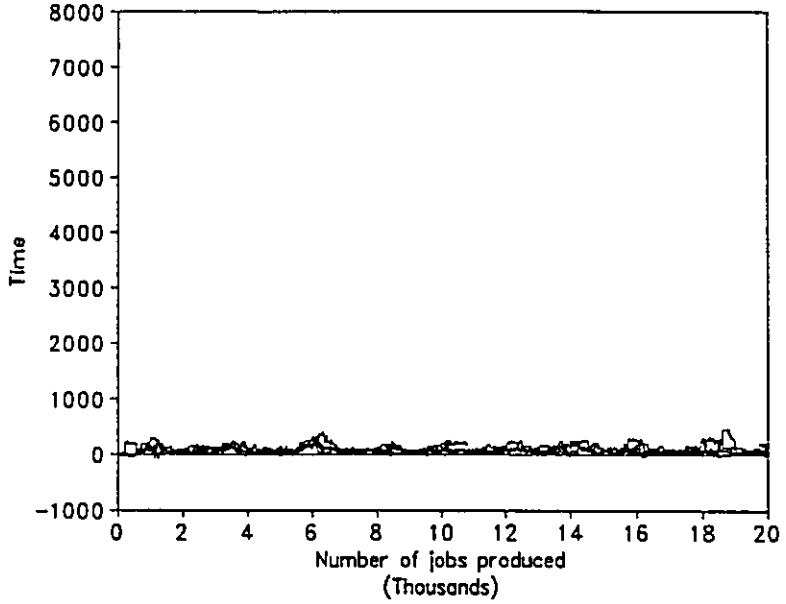


Figure E.2 E_{10}_shifted^$_i$ i=4..7, vs. number of jobs after applying the MCT algorithm MF = 0.25

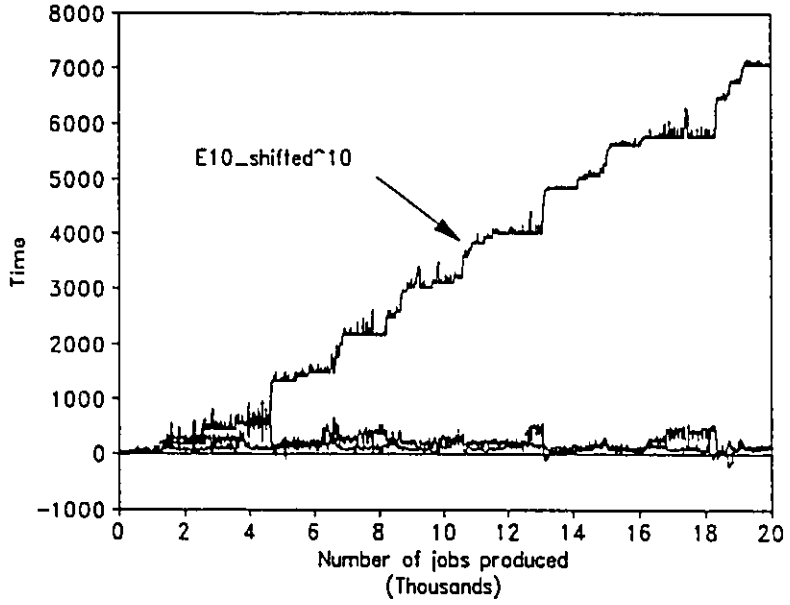


Figure E.3 E_{10}_shifted^$_i$ i=8..10, vs. number of jobs after applying the MCT algorithm MF = 0.25

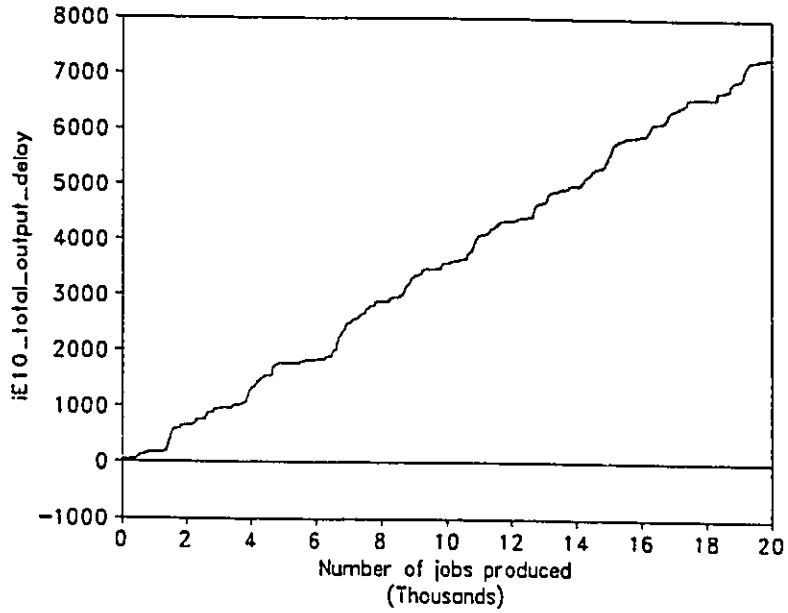


Figure E.4 iE_{10}_total_output_delay vs. number of jobs after applying the MCT algorithm MF = 0.25

	Mean	Std. dev.	Minimum	Maximum
$Stage_1$	53.59	20.45	1.51	115.19
$Stage_2$	53.65	20.51	1.45	117.78
$Stage_3$	53.47	20.36	2.47	120.42
$Stage_4$	53.63	20.52	3.19	124.60
$Stage_5$	53.61	20.48	2.13	127.21
$Stage_6$	53.65	20.48	1.71	127.63
$Stage_7$	53.61	20.45	2.82	124.80
$Stage_8$	53.56	20.55	2.06	133.90
$Stage_9$	53.56	20.50	2.24	159.31
$Stage_{10}$	53.56	20.42	3.58	137.51
Average	53.59	20.47	2.32	128.84

Table E.1 Cycle time statistics after applying the MCT algorithm, MF = 0.25

Actual Production Time = 1199346.5

Error (percentage of expected production time) = 0.054%

Efficiency = 89.37%

APPENDIX F

System Output 'Shifted' Plots after applying the maximum cycle time 'algorithm'

MF = 0 (the preliminary criteria)

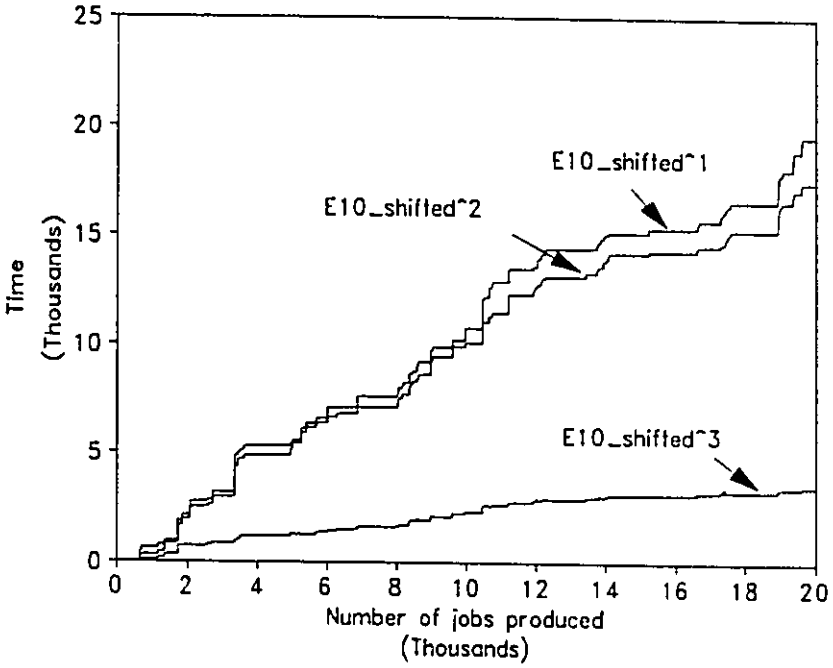


Figure F.1 E_{10}_shifted^$_i$ i=1..3, vs. number of jobs after applying the MCT algorithm MF = 0.00

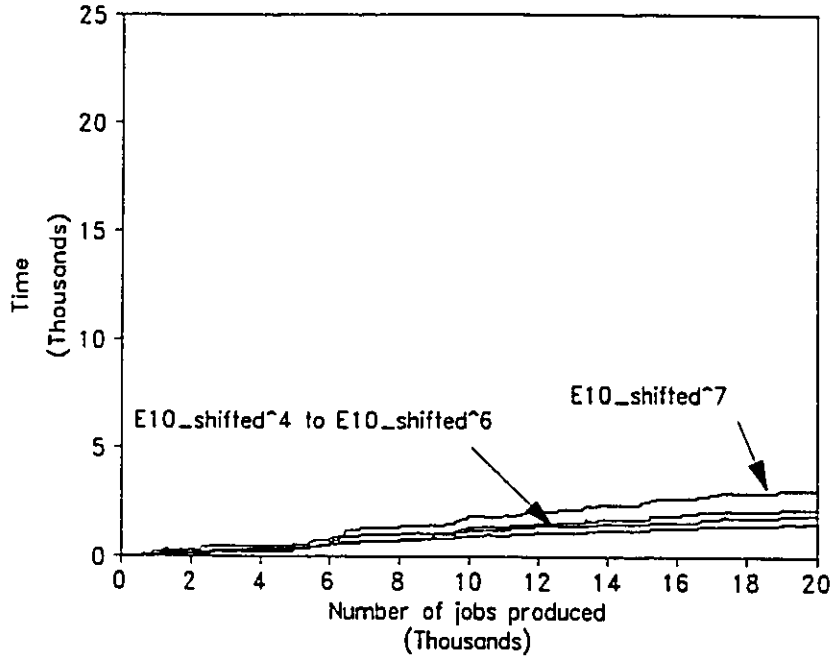


Figure F.2 E_{10}_shifted^$_i$ i=4..7, vs. number of jobs after applying the MCT algorithm MF = 0.00

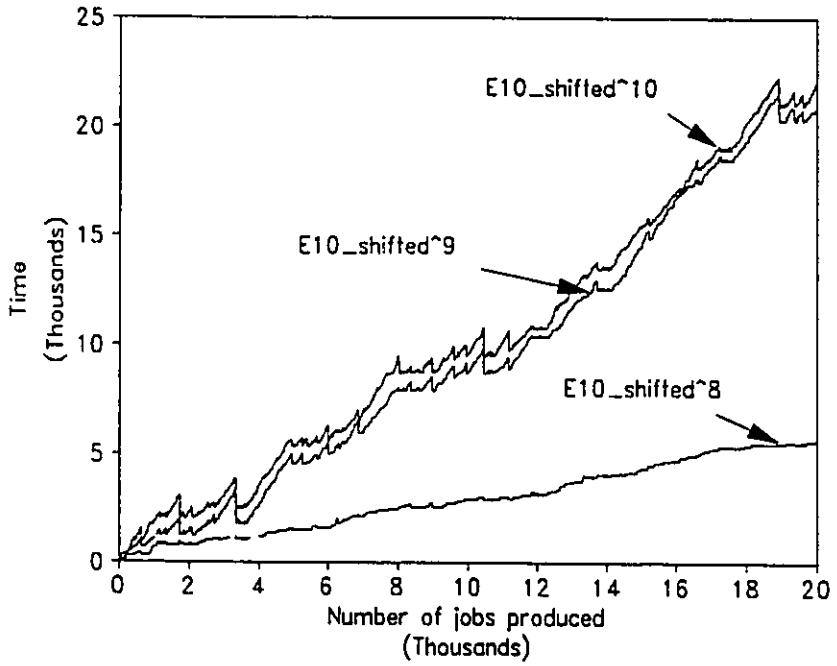


Figure F.3 E_{10}_shifted^$_i$ i=8..10, vs. number of jobs after applying the MCT algorithm MF = 0.00

	Mean	Std. dev.	Minimum	Maximum
$Stage_1$	53.22	19.93	1.51	90
$Stage_2$	48.31	14.05	1.45	60.00
$Stage_3$	48.23	13.98	2.47	60.00
$Stage_4$	48.29	14.01	3.19	60.00
$Stage_5$	48.30	14.01	2.13	60.00
$Stage_6$	48.32	13.97	1.71	60.00
$Stage_7$	48.32	14.05	2.82	60.00
$Stage_8$	48.22	14.05	2.06	60.00
$Stage_9$	48.26	14.04	2.24	79.12
$Stage_{10}$	53.20	19.85	3.58	90
Average	49.27	15.20	2.32	67.91

Table F.1 Cycle time statistics after applying the MCT algorithm, MF = 0.00

Actual Production Time = 1102399.9

Error (percentage of expected production time) = 8.13%

Efficiency = 89.38%

APPENDIX G

Figures for the Preliminary Study of Buffer Allocation

	Mean	Std. dev.	Minimum	Maximum
case 1	59.98	59.97	0.0063	587.95
case 2	59.93	42.39	0.35	379.32
case 3	59.70	34.36	0.47	292.07
case 4	60.03	29.96	1.25	271.81
case 5	59.88	26.73	5.60	243.16
case 6	59.88	18.87	11.01	167.11
case 7	60.01	13.41	17.45	135.35
case 8	60.03	10.96	25.75	115.71
case 9	59.93	7.71	34.62	101.66

Table G.1 Cycle time sets with same mean and different deviation for generating figure 7.9 and figure G.2

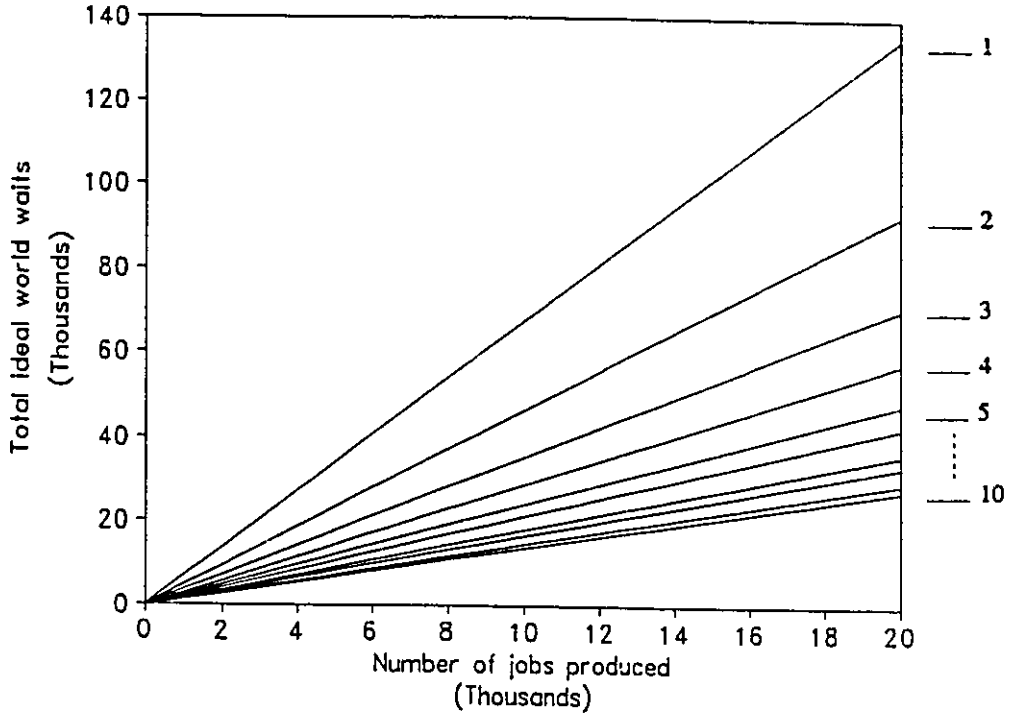


Figure G.1 Ideal world total waits vs. different storage sizes

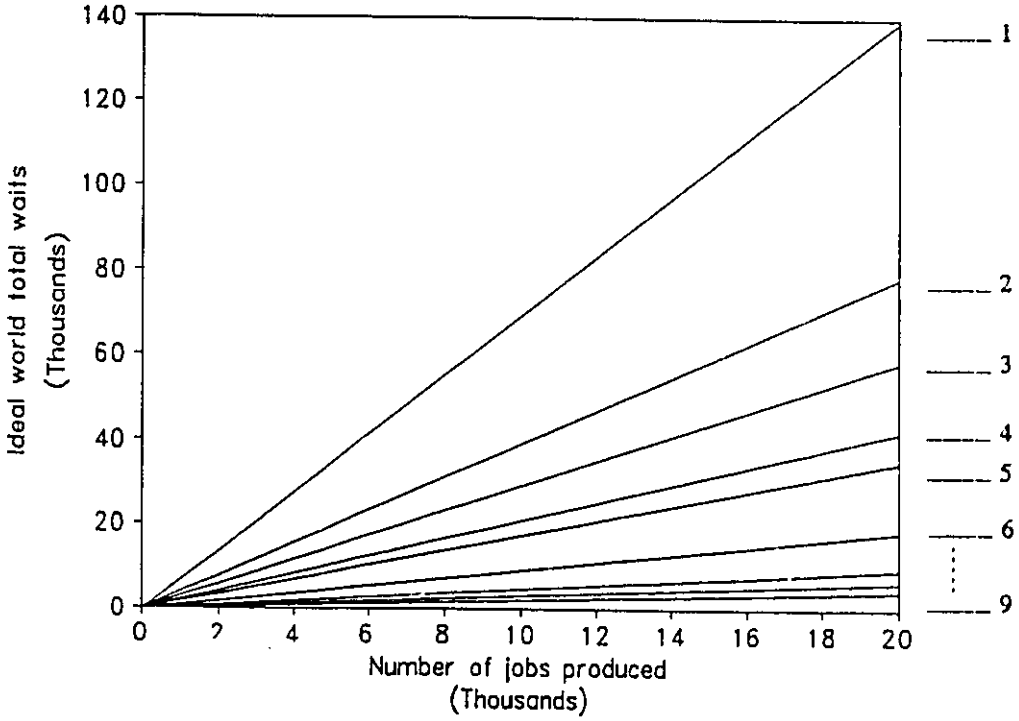


Figure G.2 Ideal world total waits vs. different cycle time distributions

APPENDIX H

Figures for the reversibility property example

	Mean	Std. dev.	Minimum	Maximum
$Stage_1$	55.92	27.88	1.41	274.71
$Stage_2$	60.02	29.82	1.45	250.85
$Stage_3$	63.91	31.87	1.61	313.96

Table H.1 Cycle time statistics for the three stages in the example of reversibility

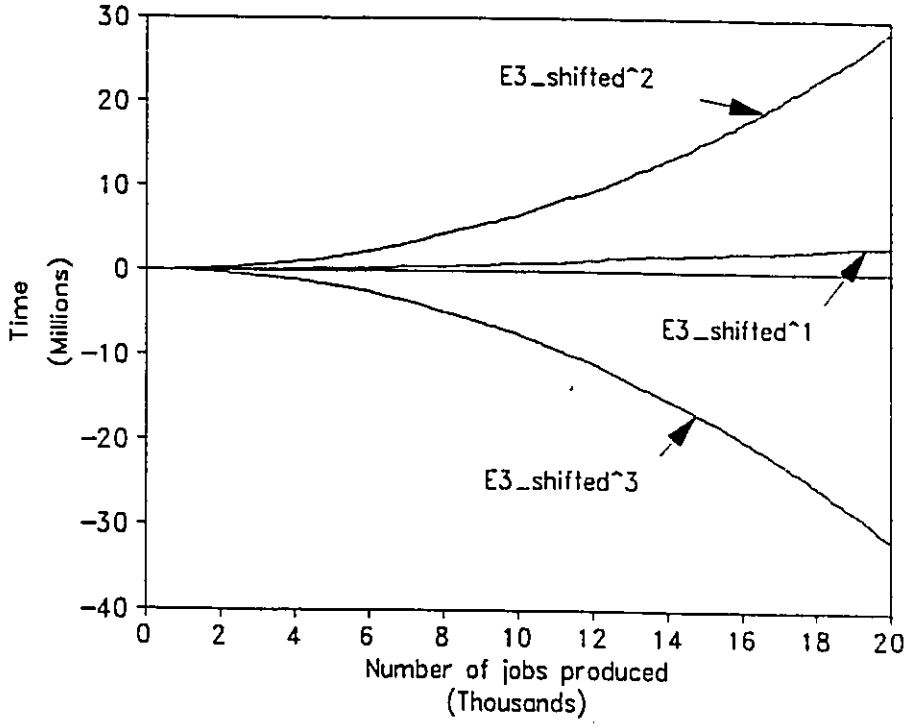


Figure H.1 Itemized E_3_shifted^

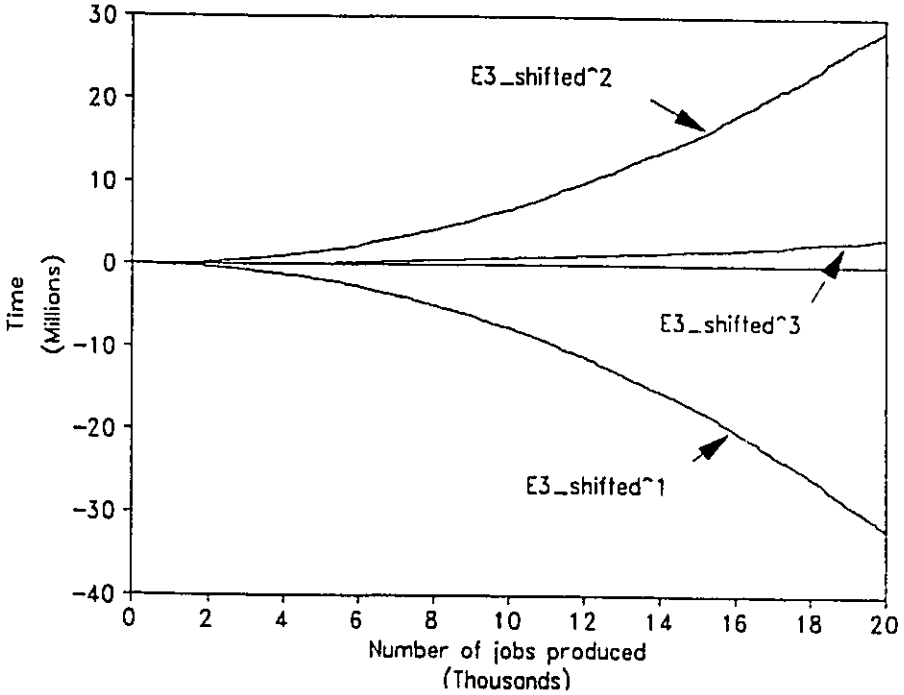


Figure H.2 Itemized E_3_shifted^, with $stage_1$ and $stage_3$ interchanged

REFERENCES

Anderson, D. R., Moodie, C. L., "Optimal Buffer Storage Capacity in Production Line Systems", International Journal of Production Research, 7, 233-240, 1969.

Barten, K. A., "A Queuing Simulator for Determining Optimum Inventory Levels in a Sequential Process", Journal of Industrial Engineering, 13, 245-252, 1962.

Blumenfeld, D. E., "Performance Comparison of Assembly Systems with Fixed and Flexible Cycle Times", The International Journal of Flexible Manufacturing Systems, 1, 207-222, 1989.

Buxey, G. M., Slack, N. D., Wild, R., "Production Flow Line Systems Design - A Review", AIIE Transactions, March, 1973.

Buzacott, J. A., "Automatic Transfer Lines with Buffer Stocks", International Journal of Production Research, 5, 3, 183-200, 1967.

Buzacott, J. A., "The Role of Inventory Banks in Flow-line Production Systems", International Journal of Production Research, 9, 4, 425-436, 1971.

Buzacott, J. A., "The Effect of Station Breakdown and Random Processing Times on the Capacity of Flow Lines with In-Process Inventory", AIIE Transactions, IV, 4, 308-312, 1972.

Buzacott, J. A., Hanifin, L. E., "Models of Automatic Transfer Lines with Inventory Banks - A Review and Comparison", AIIE Transactions, 197-207, June, 1978.

Buzacott, J. A., "Abandoning the Moving Assembly Line: Models of Human Operators and Job Sequencing", International Journal of Production Research, 28, 5, 821-839, 1990.

Dallery, Y., David, R., Xie, X. L., "An Efficient Algorithm for Analysis of Transfer Lines with Unreliable Machines and Finite Buffers", IEE Transactions, 20, 3, 280-283, Sep, 1988.

Gershwin, S., Berman, O. "Analysis of Transfer Lines Consisting of Two Unreliable Machines with Random Processing Times and Finite Storage Buffers", AIIE TRANSACTIONS, 13,1, 2-11, Mar, 1981.

Gershwin, S. B., Schick, I. C., "Modelling and Analysis of Three-Stage Transfer Lines with Unreliable Machines and Finite Buffers", Operations Research, 31, 2, 354-380, Mar, 1983.

Gershwin, S. B., "An Efficient Decomposition Method for the Approximate Evaluation of Tandem Queues with Finite Storage Space and Blocking", Operations Research, 35, 2, 291-305, Mar, 1987.

Gross, D., Harris, C. M., "Fundamentals of Queueing Theory", Second Edition, John Wiley & Sons, 1985.

Hatcher, J. M., "The Effect of Internal Storage on the Production Rate of a Series of Stages Having Exponential Service Times", AIIE Transactions, 1, 150-156, 1969.

Hillier, F. S., Boling, R. W., "The Effect of Some Design Factors on the Efficiency of Production Lines with Variable Operation Times", Journal of Industrial Engineering, 17, 12, 651-658, 1966.

Hillier, F. S., So, K. C., "The Effect of Machine Breakdowns and Inter-stage Storage on the Performance of Production Line Systems", International Journal of Production Research, 29, 10, 2043-2055, 1991.

Hira, D. S., Pandey, P. C., "Efficiency of Manual Flow Line Systems - Predictive Equations", International Journal of Production Research, 25, 4, 603-614, Aug, 1986.

Huang, P., Y. Houck, B. L. W., "A Cellular Manufacturing: An Overview and Bibliography", Production and Inventory Management, Fourth Quarter, 83-92, 1985.

Hunt, G. C., "Sequential Arrays of Waiting Lines", Operations Research, 4, 6, 674-683, 1956.

Ignall, E., Silver, A., "The Output of A Two-Stage System with Unreliable Machines and Limited Storage", AIIE Transactions, 183-169, June, 1977.

Kraemer, S. A., Love, R. F., "A Model for Optimizing the Buffer Inventory Storage Size in a Sequential Production System", AIIE Transactions, 2, 1, 64-69, 1970.

Marks, N. B., "Output Analysis of an Automated Serial Assembly Line", International Journal of Production Research, 25, 8, 1197-1207, 1987.

Masso, J., Smith, M. L., "Inter-stage Storage for Three Stage Lines Subject to Stochastic Failures", AIIE Transactions, 6 4, 354-358, Dec, 1974.Murphy, R. A., "Estimating the Output of a Series Production System AIIE Transactions" 10, 2, 139-148, 1978.

Muth, E. J., "The Production Rate of a Series of Work Stations with Variable Service Times", International Journal of Production Research, 11, 2, 155-169, 1973.

Perros, H. G. "Queueing Networks with Blockings: A Bibliography"

Van Ryzin, G., Lou, S. X. C., Gershwin, S. B., "Production Control for a Tandem Two-Machine System", IIE Transactions, 25, 5, 5-20, Sep, 1993.

Sheskin, T. J., "Allocation of Inter-stage Storage along an Automatic Production Line", AIIE Transactions, 8, 146-152, 1976.

Soyster, A. L., Schmidt, J. W., Rohrer, M. W., "Allocation of Buffer Capacities for a Class of Fixed Cycle Production Lines", AIIE Transactions, 11, 140-146, 1979.

Vasil'ev, V. N., Dashchenko, A. I., Redin, V. N., "Assembly-Line Output: Problems and Approaches", International Journal of Advance Manufacturing Technology, 4, 384-397, 1989.

BIBLIOGRAPHY

Akella, R., Choong, Y., Gershwin, S. B., "Performance of Hierarchical Production Scheduling Policy", IEEE Transactions on Components, Hybrids, and Manufacturing Technology, CHMT-7, 3, 225-240, 1984.

Akella, R., Kumar, P. R., "Optimal Control of Production Rate in a Failure Prone Manufacturing System", IEEE Transactions on Automatic Control, AC-31, 2, 116-126, Feb, 1986.

Altiok, T., Stidham, S., "A Note On Transfer Lines With Unreliable Machines, Random Processing Times, and Finite Buffers" IIE Transaction, 125-127, June, 1981.

Altiok, T., "Approximate Analysis of Exponential Tandem Queues with Blocking", European Journal of Operations Research, 11, 390-398, 1982.

Altiok, T., "Approximate Analysis of Queues in Series with Phase-Type Service Times and Blocking", Operations Research, 37, 4, 601-610, July, 1989.

Brandwajn, A., Jow, Y. L., "Approximation Method for Tandem Queues with Blocking", Operations Research, 36, 1, 73-83, Jan, 1988.

Buffa, E. S., "Pacing Effects in Production Lines", Journal of Industrial Engineering, 12, 6, 383-386, Nov, 1961.

Burke, P. J., "The Output of a Queueing System", Operations Research, V 4, No. 6, 699-704, 1956.

Buzacott, J. A., "Prediction of the Efficiency of Production Systems Without International Storage", International Journal of Production Research, 6, 173-188, 1968.

Caseau, P., Pujolle, G., "Throughput capacity of a Sequence of Queues with Blocking Due to Finite Waiting Room", IEEE Transactions of Software Engineering, SE-5, 6, 631-642, Nov, 1979.

Cochran, J. K., Lin, L, "Compound Dynamic Event Metamodels for Electronic Assembly Line Systems", IIE transactions, 25, 2, 12-25, March, 1993.

Commault, C., Dallery, Y., "Production Rate of Transfer Lines Without Buffer Storage", IIE Transactions, 22, 4, 315-329, 1990.

D'Angelo, H., Caramanis, M., Finger, S., "Event-Driven Model of Unreliable Production Lines with Storage", International Journal of Production Research, 26, 7, 1173-1182, July, 1987.

De Koster, M. B. M., "Estimation of Line Efficiency by Aggregation", International Journal of Production Research, 25, 4, 615-626, 1987.

De Koster, M. B. M., "Approximation of Flow Lines with Integrally Controlled Buffers", IIE Transactions, 20, 4, 374-381, Dec, 1988.

De Koster, M. B. M., "An Improved Algorithm to Approximate the Behaviour of Flow Lines", International Journal of Production Research, 26, 4, 691-700, 1988.

Disney, R. L., "Random Flow in Queueing Networks", AIIE Transactions, 7, 3, Sept, 1975.

Evans, R. V., "Capacity of Queueing Networks", Operations Research, 15, 530-536, 1967.

Freeman, M. C., "The Effects of Breakdowns and Inter-stage Storage on Production Line Capacity", Journal of Industrial Engineering, 15 4, 194-200, July, 1964.

Fry, T. D., Cox, J. F., "Manufacturing Performance: Local Versus Global Measures", Production and Inventory Management Journal - Second Quarter, 52-56, 1989.

Gershwin, S. B., Akella, R. Choong, Y. F., "Short-term Production Scheduling of an Automated Manufacturing Facility", IBM Journal of Research Development, 29, 4, 392-400, July, 1985.

Goode, Henry, P., Saltzman, S., "Estimating Inventory Limits in a Station Grouped Production Line", Journal of Industrial Engineering, 13 6, 485-490, Nov, 1962.

Gordon, W. J., Newell, G. F., "Cyclic Queueing Systems with Restricted Length Queues", Operations Research, 15, 266-278, 1967.

Hildebrand, D. K., "On the Capacity of Tandem Server, Finite Queue, Service Systems", Operations Research, 16, 72-82, 1968.

Hillier, F., Boling, R., "Finite Queues in Series with Exponential or Erlang Service Times-A Numerical Approach", Operations Research 15, 286-303, 1967.

Ho, Y. C., Eyler, M. A., Chien, T. T., "A Gradient Technique for General Buffer Storage Design in a Production Line", International Journal of Production Research, v. 17, no. 6, 557-580, 1979.

Hong, Y., Glassey, C. R., Seong, D., "The Analysis of a Production Line with Unreliable Machines and Random Processing Times", IIE Transactions, 24, 1, 77-83, Mar, 1992.

Hopp, W. J., Spearman, M. L., "Throughput of a Constant Work in Process Manufacturing Line Subject to Failures", International Journal of Production Research, 29, 3, 635-655, 1991.

Huang, P. Y., Chen, C. S., "Flexible Manufacturing Systems: An Overview and Bibliography", Production and Inventory Management, Third Quarter, 80-90, 1986.

Iyama, T., Ito, S., "The Maximum Production Rate for an Unbalanced Multi-Server Flow Line System with Finite Buffer Storage", International Journal of Production Research, 25, 8, 1157-1170, 1987.

Jackson, R. R. P., "Queueing Systems with Phase Type Service", Operations Research Quarterly, V 5, 1954.

Johri, P. K., "Engineering a Circuit Board Assembly Line for a Desired Capacity and Flowtime", Journal of Manufacturing Systems, 10, 6, 492-500,

Johri, P. K., "A Linear Programming Approach to Capacity Estimation of Automated Production Lines with Finite Buffers", International Journal of Production Research, 25, 6, 851-866, 1987.

Johri, P. K., "Iterative Algorithms for Estimating the Capacity of Automated Production Lines with Finite Buffers", International Journal of Production Research, 27, 4, 687-699, Feb, 1988.

Kimemia, J., Gershwin, S. B., "An Algorithm for the Computer Control of a Flexible Manufacturing System", IIE Transactions, 353-362, Dec, 1983.

Klastorin, T. D., Matheson, L. Moinzadeh, K., "On The Use of Buffer Inventories to Minimize Costs with an Unreliable Machine", IIE Transactions, 25, 5, 50-62, September, 1993.

Law, S. S., "A Statistical Analysis of System Parameters in Automatic Transfer Lines", International Journal of Production Research, v. 19, no. 6, 709-724, 1981.

Lou, S. X. C., Kager, P. W., "A Robust Production Control Policy for VLSI Wafer Fabrication", IEEE Transactions on Semiconductor Manufacturing, 2, 4, 159-164, Nov, 1989.

Mascolo, M. D., David, R. Dallery, Y., "Modelling and Analysis of Assembly Systems with Unreliable Machines and Finite Buffers", IIE Transactions, 23, 4, 315-330, Dec, 1991.

McCormick, S. T., Pinedo, M. L., Shenker, S. Wolf, B., "Sequencing in an Assembly Line with Blocking to Minimize Cycle Time", Operations Research, 37, 6, 925-935, Nov, 1989.

Murphy, R. A., "The Effect of Surge on System Availability", AIIE Transactions, 7, 4, 439-443, 1975.

Ohmi, T., "An Approximation for the Production Efficiency of Automated Transfer Lines with In-Process Storage", AIIE Transactions, 13, 22-28, 1981.

Ohno, K., Ichiki, K., "Computing Optimal Policies for controlled Tandem Queueing Systems", Operations Research, 35, 121-126, Apr, 1986.

Okamura, K., Yamashi, H., "Analysis of the Effect of Buffer Storage Capacity in Transfer Line Systems", AIIE Transactions 9, 2, 127-135, 1977.

Park, T. , Steudel, H. J., "A Model for Determining Job Throughput Times for Manufacturing Flow Line Workcells with Finite Buffers", International Journal of Production Research, 29, 10, 2025-2041, 1991.

Perros, H. G., "Queueing Networks with Blocking: A Bibliography", Performance Evaluation Review, 12, 2, 8-12, 1984.

Pinedo, M., Wolff, R. W., "A Comparison between Tandem Queues with Dependent and Independent Service Times", Operations Research, 30, 464-479, 1982.

Rajagopalan, A., "Flexible Versus Dedicated Technology: A Capacity Expansion Model", The International Journal of Flexible Manufacturing Systems, 5, 129-142, 1993.

Rao, N. P., "Two-Stage Production Systems with Intermediate Storage", AIIE Transactions, 7 4, 414-421, Apr, 1974.

Rao, N. P., "On the Mean Production Rate of a Two-Stage Production System of the Tandem Type", International Journal of Production Research, 13 2, 207-217, 1975.

Russel, D. W., "Production Monitoring - A Proven Productivity Tool", International Journal of Advance Manufacturing Technology, 4, 335-348, 1989.

So, K. C., "Allocation Buffer Storages in a Flexible Manufacturing System", The International Journal of Flexible Manufacturing Systems, 1, 223-237, 1989.

Takahashi, Y., Miyahara, H. Hasegawa, T., "An Approximation Method for Open Restricted Queueing Networks", Operations Research, 28, 594-602, 1980.

Tempelmeier, H., Kuhn, H. Tetzlaff, U., "Performance Evaluation of Flexible Manufacturing Systems with Blocking", International Journal of Production Research, 27, 11, 1963-1979, 1989.

Tsiotras, G., Badr, H., Beltrami, E., "Blocking Probabilities for a Class of Two-Station Tandem Queueing Models", IEE Transactions, 23, 4, 331-345, Dec, 1991.

Wijngaard, J., "The Effect of Inter-stage Buffer Storage on the Output of Two Unreliable Production Units in Series, with Different Production Rates", AIIE Transactions 11, 1, 42-47, 1979.

Xie, X, "Performance Analysis of a Transfer Line with Unreliable Machines and Finite Buffers", IIE Transactions, 25, 1, 99-108, January, 1993